TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JUHA KOLJONEN

AUTOMATIC MODEL–BASED PID TUNING OF A SERVO AXIS

Master of Science Thesis

Examiners: Dr. Niko Siltala and
Associate Professor Minna Lanz

The examiner and topic of the thesis
were approved on 2nd May 2018

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Mechanical Engineering
JUHA KOLJONEN: Automatic Model–based PID Tuning of a Servo Axis
Master of Science Thesis, 71 pages, 7 Appendix pages
August 2018
Major: Integrated Production Technology and Production
Examiners: Dr. Niko Siltala and Associate Professor Minna Lanz
Keywords: model–based, PID, PI, controller, tuning, servo, robot, ISE, performance

This thesis considers the automatic parameter tuning of a servo axis. The target servo axis of this thesis is controlled with an industrial servo drive that utilizes PID controllers in its internal structure. In particular, the focus of this study is on the functional behavior and performance of common controller auto-tuning methods when used in conjunction with the OptoFidelity OptoDrive servo drive. The purpose of this thesis is to address the state of robot setup methodology where a trained professional is responsible for the manual tuning of each servo axis, and to build a basis for PID auto-tuning with the OptoDrive. Manual tuning takes a considerable amount of time and has proven to be problematic in large scale applications, such as mass-production environments. The research effort on auto-tuning methods in this thesis is motivated by the desire to eliminate the need of manual controller tuning altogether.

The initial results suggest that automatic controller parameter tuning is feasible for the OptoDrive's axis velocity and position controllers. Four controller auto-tuning methods were tested, from which setpoint overshoot, closed-loop SIMC, and Ziegler-Nichols methods were found to be suitable for tuning the velocity controller of the OptoDrive. The Ziegler-Nichols was found to be the only method that is suitable for tuning the position controller of the OptoDrive. Therefore, the Ziegler-Nichols auto-tuning method was studied further, and effort was put into automatically optimizing the mathematical formulae utilized by the method to achieve the best possible auto-tuning results. After optimization, it produced the best performance results recorded in the experiments of this thesis. The performance surpassed the average tuning performance of humans by significant margins, and even more importantly, with a significantly better standard deviation of performance.

The tests were conducted on a single linear servo axis, whose mass was varied in three configurations to emulate three systems with different inertia. The results obtained in this thesis suggest that automatic tuning should be implemented when the best possible robot performance is desired with the OptoDrive. If the desire is to use an other servo drive than the OptoDrive, the performance tests should be rerun to be valid. As for the impact of this thesis, a patenting process on the aforementioned optimization method is ongoing, and OptoFidelity is moving to use PID auto-tuning with the OptoDrive.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Konetekniikan DI-tutkinto-ohjelma
JUHA KOLJONEN: Servoakselin PID-säätimen automaattinen mallipohjainen viritys
Diplomityö, 71 sivua, 7 liitesivua
Elokuu 2018
Pääaine: Integroitu tuotekehitys ja tuotanto
Tarkastajat: Tutkijatohtori Niko Siltala ja Apulaisprofessori Minna Lanz
Avainsanat: mallipohjainen, PID, PI, säädin, viritys, servo, robotti, ISE, suorituskyky

Työ käsittelee servoakselin ohjaimen parametrien automaattista viritystä. Mittauksissa käytetyn servoakselin ohjain, OptoFidelity OptoDrive, on toteutettu PID-säätimillä. Työ keskittyy yleisten automaattisten PID-säätimen viritysmenetelmien toimintaan ja suorituskykyyn OptoDriven kanssa käytettynä. Nykytilassa OptoDriven parametrit viritetään manuaalisesti. Työn tarkoitus on luoda pohja OptoDriven automaattiselle viritykselle, sillä manuaalinen PID-säätimien viritys on hidasta. Hitautensa vuoksi se on osoittautunut ongelmalliseksi erityisesti massatuotannossa.

Työn tulosten mukaan OptoDriven säätimien automaattinen viritys on mahdollista. Neljästä kokeillusta menetelmästä setpoint overshoot, closed-loop SIMC ja Ziegler-Nichols -menetelmät havaittiin soveltuviksi OptoDriven nopeussäätimen viritykseen. Menetelmistä ainoastaan Ziegler-Nichols havaittiin soveltuvaksi OptoDriven paikkasäätimen viritykseen. Ziegler-Nicholsin optimointin kehitettiin geneettiseen algoritmiin perustuva menetelmä. Optimoitu Ziegler-Nichols tuotti OptoDrivelle viritysparametrit, jotka tuottivat parhaat työssä mitatut suorituskykytulokset. Mitattu suorituskyky on merkittävästi ihmisten tuottamien säätöparametrien suorituskykyä parempi.

Testus suoritettiin lineaariservoakselilla, jonka kelkan massaa muunneltiin massalevyillä. Työn tulokset osoittavat, että parhaan suorituskyvyn saavuttamiseksi OptoDriven säätimet tulisi virittää automaattisesti. Mikäli työn tuloksia halutaan soveltaa muille servo-ohjaimille, työn mittaukset tulee uusia tapauskohtaisesti. Työn aikana kehitetty automaattisen viritysmenetelmän optimointimenetelmä patentoidaan. OptoFidelity ottaa käyttöön työkalun, jolla OptoDrive voidaan virittää automaattisesti.

PREFACE

The research and testing were conducted at OptoFidelity Oy, in Tampere, Finland. Practical research and insight for the background of this thesis was gathered at OptoFidelity premises in Finland, USA, and China. I would like to gratefully acknowledge OptoFidelity Oy for providing me the opportunity to work with world class robotics, and for the possibility to carry out my thesis while working full-time as a HW engineer.

I gratefully acknowledge Associate Professor Minna Lanz and Doctor Niko Siltala for supervising my thesis. Your guidance has given me a great basis for analytical thinking and highly useful methods for researching the topics discussed in this thesis.

A warm thank you to Hans Kuosmanen, SVP of OptoFidelity Oy, for supervising and supporting my work at OptoFidelity. The discussions we have had during the tuning method development and testing process have given me great insights and motivation to research the topic. Furthermore, without the support and time management efforts by Hans, I probably would not have finished this thesis in the current schedule. Thank you.

I want to thank Eero Heinänen for his support during my thesis process and for great insights in building the software implementations used in this thesis. The platform, co-developed with Eero, has been instrumental in handling the high-performance inputs, outputs and state logic required for this research. The KH-method we developed has proven itself to yield great results in servo drive tuning. Figuring out the ins and outs of a servo drive system was not only straightforward, but also fun with you. Thank you.

I also want to thank my family for support during the writing process, Kyle for giving great advice on English language structure and spelling, and Janne for tips and opinions regarding the structure and literary style of a thesis. For the anonymous experts in OptoFidelity that participated in the manual tuning test, thank you, your input has been very important. Thank you Joonas for your advice regarding the thesis process and for mental support. Let's make sure that the sauna is hot, and the beer is cold.

Tampere, 1st August 2018

Juha Koljonen

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SYMBOLS AND ABBREVIATIONS

AGA	Adaptive genetic algorithm, (optimization method)
CSV	Comma separated-values, (file/data format)
EMF	Electromotive force, (electromagnetic characteristic)
FLC	Fuzzy logic controller, (controller)
FOPDT	First order plus dead time, (process model)
FTT	Fast Fourier transform, (signal conversion method)
IAE	Integrated absolute error, (performance index)
IMC	Internal model control, (auto-tuning method)
ISE	Integrated squared error, (performance index)
ITAE	Integrated time-weighted absolute error, (performance index)
ITSE	Integrated time-weighted squared error, (performance index)
PFC	Predictive functional controller, (controller)
PID	Proportional integral derivative, (controller)
SIMC	Simple internal model control, (auto-tuning method)
SISO	Single input single output, (controller type)

σ	Standard deviation
τ_c	Process time constant
τ_t	Adjustment parameter
θ_d	Process dead time
A	Adjusted amplitude
A_o	Peak amplitude (relative)
b	Relative process steady state change
ΔU	Relative change in controller output
Δy	Relative change in process value
e	Process error
K_{d}	Derivative gain
K_{i}	Integral gain
$K_{\mathrm{i}p}$	Integral gain of a parallel controller
$K_{\mathrm{i}i}$	Integral gain of an ideal controller
K_{p}	Proportional gain
K_{p0}	In-experiment proportional gain
K_{pg}	Process gain
$K_{\mathrm{p}i}$	Proportional gain of an ideal controller
$K_{\mathrm{p}p}$	Proportional gain of a parallel controller
$K_{\mathrm{p}u}$	Ultimate proportional gain

P_{osc}	Oscillation period
$P(s)$	Process function
r	Process setpoint
t	Time
T_d	Derivative time
T_i	Integral time
t_p	Process peak time
U	Controller output
u_d	Derivative action
U_i	Output of an ideal controller
u_i	Integral action
U_p	Output of a parallel controller
u_p	Proportional action
y	Process value
Δy_∞	Process steady state change
y_m	Measured process value
Δy_p	Peak process change
Δy_s	Applied setpoint change

1. INTRODUCTION

The increasing complexity of electronic devices has lead to a growing demand for various applications of test robotics. When considering the field of test robotics, the robot manufacturers should be able to adapt modularized robot products to a vast number of test applications. This is to reach the demanding project timelines, and to keep the development costs from getting out of hand. Additionally, the robots are required to have a long lifetime to provide a sufficient return on investment for the customers.

When adapting robots for different applications, the dynamical properties of the systems change. Manufacturing tolerances and normal wear have a further affect on the robots. This causes a batch of robots to have different dynamical properties between them, which requires actions from the robot controller's standpoint to keep the control system working optimally. The controller's performance has a significant impact on the performance of the final test robot.

Robot controllers typically utilize proportional integral derivative (PID) controllers to control the motion of the robots' axes. The dynamical properties of the robots and their individual axes are reflected by the PID controller's tuning parameters. Thus, the tuning parameters must be modified for new robot designs and even individual mass-produced robots if high functioning precision and accuracy of the robots is required. In an optimal case, the tuning parameters would be adjusted periodically as one step of the robots' maintenance procedure. The use of sub-optimal tuning parameters causes sub-optimal controller behavior. It is detrimental to the robots' cross-track error, which measures error in a robot's position during multi-axis synchronous moves. In practice, sub-optimal controller behavior causes a robot's multi-axis linear moves to look like a banana or an s-curve.

The motivation for this thesis stems from the fact that tuning the PID controller's parameters properly by hand is time consuming and difficult. PID parameter tuning requires skilled personnel, whose expertise could be better used elsewhere. When it comes to fine tuning the PID controller's tuning parameters for individual robots in a mass production phase, manual tuning is effectively impossible due to time and resourcing constraints. Therefore, an automatic PID parameter tuning method is desirable, as it has the opportunity to eliminate the need of manually

tuning the robot controllers. It also allows optimization of robot controllers' tuning parameters throughout the robots' lifetime, which might help prevent a decay in robots' performance as they age and are subjected to mechanical wear.

While some commercial motor controller models are equipped with auto-tuning functions, the OptoDrive motor controller that is currently used in OptoFidelity's test robots has been developed in house to increase the amount of vertical integration of the robotic systems, and is missing the auto-tuning feature. The performance and, most importantly, the repeatability of such an auto-tuning function should be well tested before practical use. Experience shows that reaching the required micrometre-scale precision and accuracy with a test robot is a difficult task that requires highly optimized PID controller tuning parameters, and often multiple rounds of manual tuning. Currently there is no existing knowledge about auto-tuning of the OptoDrive. It has never been tested. Due to the difficulty of manual tuning, obtaining good performance with auto-tuning should not be taken for granted. Additionally, the auto-tuning methods' functionality might not be suitable for the OptoDrive or their behavior might be incompatible with measurement robots.

1.1 Objectives of the thesis

Numerous automatic tuning methods have been suggested in literature for PID controllers. Industrial servo drives often implement auto-tuning methods, whose underlying technical solutions are not publicly available. A two-step approach has been taken into auto-tuning the OptoDrive. The first step is to produce a well performing controller tuning parameter set, which should be able to move the servo axis between position setpoints. The second tuning step will iteratively fine-tune the controller tuning parameters using machine learning algorithms to achieve high functional precision of the robot. The primary focus of this thesis will be on implementing the first step. A master's thesis focusing on the second step is being done parallel to this thesis by Eero Heinänen, an OptoFidelity software engineer.

The focus of this thesis is on model–based auto-tuning methods, as opposed to model–free methods. The model–based methods utilize some form of a test or an experiment to find a mathematical representation of a target system, from which suitable controller tuning parameters can be calculated for a target system. These kinds of methods can be used to obtain a set of controller tuning parameters quickly, whereas model–free methods are generally based on iterative optimization, taking several hours. Optimization methods work faster when there are already some initial parameters, so it is logical to use a model–based method in the first OptoDrive tuning step.

The primary goal of this thesis is to find an answer to whether model–based auto-tuning is feasible for the servo position and velocity controllers of the OptoDrive. If some auto-tuning methods are found to be suitable for the controllers of the OptoDrive, their performance will be tested and compared to human tuning performance. Ideally, the best performing controller auto-tuning method could be recommended for future use after the experiments in this thesis.

1.2 Research methods and restrictions

An initial understanding of PID controller auto-tuning was established during lunch-time conversations with various TUT professors. The notes from these initial conversations and some online research became the foundation on which all the additional research was based upon. Information on PID controllers, their auto-tuning methods, and performance measurement methods were researched with literature review.

The literary review of the PID controller parameter tuning methods was started by looking into articles and publications, which compared multiple auto-tuning methods (Tan et al. 2006, Raunt and Vaishnay 2012 and Lequin et al. 2003). The articles were used to establish a basic understanding of the field. They also helped understand the basic capabilities and requirements for the use of the tuning methods. This led to the understanding, that model-finding methods of the PID controller auto-tuning methods can be partially separated from the tuning methods themselves. This is because the auto-tuning methods use only a handful of different experiments to find the system model. Conveniently, as the articles compare auto-tuning methods, they utilize controller performance measurement indices for the comparisons. This helped establish an initial understanding on the controller performance measurement, which is required to answer the research questions of this thesis.

A standardized method for measuring a system's performance was implemented to have a fair comparison of the auto-tuning methods' performance. The auto-tuning method's behavior and performance was tested on a linear motor driven axis to eliminate the effect of backlash on the measurements and general system behavior. The linear servo axis was built from spare components partially found in OptoFidelity office and partially designed and manufactured for the thesis project. Additionally, a quantitative interview was planned and conducted to find out what kind of tuning parameters humans will set on the test axis of this thesis. In the interview, OptoFidelity's engineers with prior experience on servo tuning were asked to find optimal controller tuning parameters for the linear axis. From these results, a baseline of humans' controller tuning performance could be established. Considering the two-part approach of OptoDrive's auto-tuning, discussed in Section 1.1, and the

similarity in the objectives of the two parts, the linear servo axis and the quantitative interview was shared between the two theses.

Qualitative and quantitative experimental research was conducted on the linear axis to learn about the functional behavior and performance of model-finding methods and their respective auto-tuning methods. Regarding their behavior, auto-tuning methods should be safe for the axis hardware, should not cause long periods of oscillation, and should be deterministic considering the axis positions that are reached during the tuning process. The humans' and auto-tuning methods' performance results were measured with a standardized test, using a carefully planned trajectory that is compatible with a wide variety of servo axes used by OptoFidelity. This ensures that there are no errors in the results originating from varying testing practices.

The restrictions in the scope of research and the use of its results stem from practical considerations. To improve the usability of auto-tuning, only existing servo system sensors will be used in the process. The schedule of this thesis and the selection of a linear servo axis as the test axis excludes ball-screw axes from testing. Therefore, the results of this thesis are only applicable for linear axes, driven with an OptoDrive servo drive. Their testing can be conducted based on the work that was done in this thesis, with no additional software efforts.

The velocity and position controllers of the OptoDrive are affected by coupling. If a velocity controller is misbehaving, the observed position tracking performance of the system is reduced, and the position controller needs to actively compensate of the errors created by the velocity controller. Additionally, if a position controller is lazy in its operation, it will produce smooth control signals to the velocity controller. This makes the velocity controller's work easier and produces better apparent performance, but reduces the total performance of the system. Because of the coupling, restrictions must be applied to keep the amount of testing in this thesis manageable. First, a good-performance position controller tuning parameter found by the OptoFidelity's experts shall be used at all times, when auto-tuning and testing the velocity controller of the OptoDrive. Secondly, the best velocity controller tuning parameters found by the means of auto-tuning will be used when auto-tuning and testing the position controller of the OptoDrive.

1.3 Outline and contribution of the thesis

The chapters along with their main contents are summarized below. This thesis is divided into eight chapters. It first considers the required theory, then the method of testing, and finally the actual auto-tuning methods' behavior and performance.

Chapter 2 presents the theory and mathematical basis for this thesis, found in the literature review. PID controllers' individual elements and relevant controller variants compiled from the elements are presented. Auto-tuning methods found from literature and deemed interesting for testing with the OptoDrive are presented, along with the system model-finding experiments related to them. The controller performance methods along with their mathematical formulae are presented and discussed.

Chapter 3 is an introduction to the test hardware and software used in this thesis. The components of the hardware are presented with their primary functions, along with an additional mass plate system. The software side of the OptoDrive and a method of collecting and streaming data from it, devised for the experiments in this thesis, is discussed.

Chapter 4 presents the standardized tests used in this thesis. A test for controller performance measurement with a given controller tuning parameter set is described. A test utilizing the performance measurement, for testing controller tuning methods, is described. The plan for finding a baseline human controller tuning performance using quantitative interview is presented.

Chapter 5 provides insight into the model-finding experiment implementations that are used by the auto-tuning methods. Three experiments were found in the literature review. The practicalities of the implementations are given for them. Relevant methods of data analysis are also discussed.

Chapter 6 contains the experimental results and their analysis. The results of the human tuning performance test are presented. The model-finding experiments for the auto-tuning methods are tested for the velocity and the position controller of the OptoDrive, with an emphasis on their behavior regarding the servo axis. Some auto-tuning methods were excluded due to the behavior of their model-finding experiments. Reasoning is given for their exclusion. The performance of suitable auto-tuning methods is measured and analysed. Finally, a method for optimizing the Ziegler-Nichols auto-tuning method, that was devised in this thesis is presented along with its performance results.

Chapter 7 discusses the results obtained in the experiments, in relation to the objectives and research questions of this thesis. The results of the experiments are summarized with discussion on the phenomena behind the results. The method for optimizing the auto-tuning methods is discussed regarding its functionality and future research that should be conducted on it. Future subjects of research are also indicated for other segments of this thesis.

Chapter 8 concludes the thesis with a summary of the results. Subjects of future research are indicated. The chapter ends with prospects of the implications of this

thesis regarding OptoFidelity's robot projects.

A suitable auto-tuning method for the OptoDrive was found in this thesis, with high enough performance to easily surpass the average performance of humans with a significantly better standard deviation of the results. For the field of PID controller auto-tuning, a method for optimizing a controller auto-tuning method was devised, with enough novelty value that it passed the novelty research conducted by a patent office. A patenting process on the optimization method is ongoing.

2. LITERATURE REVIEW OF PID CONTROLLERS

The ability to control systems and processes is the foundation for the automation of industrial systems. In general, systems and processes can be controlled in an open-loop or a closed-loop fashion. Open-loop controllers are simple structures that can react to changes in an input signal (i.e. the process setpoint). Closed-loop controllers have an additional ability to react to the state of the controlled process (i.e. the process value), thus compensating for errors in the process and giving greater control precision. Hence, the closed-loop controllers are also called feedback controllers. The control problem of 90-95 percent of processes can be satisfied with these kinds of simple single input, single output (SISO) controllers. (Ogata and Yang 2002, pp. 7-9, Koivo and Tanttu 1991, pp. 75-80)

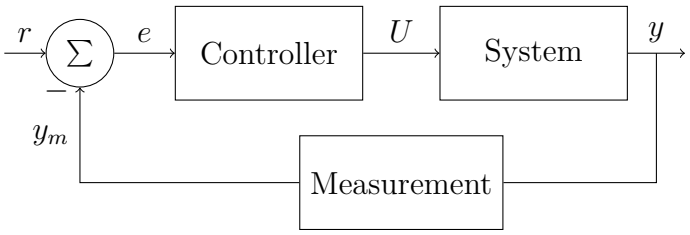

Figure 2.1 *Feedback controller*

The basic structure of a feedback controller is illustrated in Figure 2.1, where r is the process setpoint, y is the process value, e is the error between the process setpoint and the measured process value y_m, and U is the controller output. Feedback control has a multitude of applications, ranging from chemical process control to trajectory control of flight systems, whose control problems could not be satisfied with open-loop controllers. (Visioli 2006, p. 2) When applied to a digital system, a periodically executed function establishes the control action. The function reads the process value from a sensor, often called a feedback device, then calculates a value of the error variable. The control function then determines an appropriate action to the error variable using some formulae selected by the designer of the system. This action is sent to a control element (e.g. a valve, heater, electrical actuator), adjusting its operation. After a certain process-dependent dead time the change in the control element causes an effect in the controlled process. Finally, the control function is run

again after a predetermined time has elapsed. (*C2000™ Digital Controller Library* 2015, pp. 19-26)

The control problem of systems can be satisfied with many different kinds of controllers, such as the Predictive Functional Controller (PFC) (Richalet et al. 1987), the Fuzzy Logic Controller (FLC) (C.-C. Lee 1990) and the PID controller. In practice, the PID controller has established itself as the go-to controller for industrial applications thanks to its ability to control most industrial processes. (O'Dwyer 2009, pp. 1-2) With most processes, not all of the PID controller's proportional, integral and derivate elements are used. The derivative element is left out most often. Literary interest in the PID controller has been on an incline recently, thanks to the emergence of automatic controller tuning and an increased use of model predictive and thus system-adaptive control. (Åström and Hägglund 2001, p. 1163) Regarding the OptoDrive servo controller in this thesis, it only uses the proportional and integral elements of the PID controller, and therefore the derivative element can be left to only a limited level of theoretical consideration in this thesis.

This chapter will first focus on the individual control elements of the PID controller, from which full controllers are formed with different internal structures. The controller structures are important regarding the PID auto-tuning methods, as the tuning parameters are used differently in the controllers, and thus the results of the auto-tuning methods must be adapted if they are used in a different controller structure than what they were originally intended for. Conversion formulae for the tuning parameters between relevant controller structure variants are presented in this chapter. Next, common and algorithmically interesting auto-tuning methods for the PID controller are presented, along with the system model-finding experiments required for their use. Finally, PID controller performance measurement indices found in literature are presented. Their wighting of the various types of errors in the controller's operation is discussed.

2.1 PID controller elements

A key advantage of the PID controller is its simple and logical function of operation. Additionally, the PID controller is often all that is needed for satisfying easy control problems. An appropriate variant of the PID controller should be selected for each specific control problem, since the control problems where the PID controller is applied vary in nature. The variants of the PID controller are formed from individual control elements, with a logical aim to have the simplest combination to satisfy a given control problem. A final controller, that can properly respond to changes in the setpoint and the process, is formed by summing the outputs of the proportional, integral, and derivative elements together. (Visioli 2006, foreword & pp. 15-16)

As for the input of the controller elements, it is fundamentally important for a feedback controller to be able to measure the difference between the desired state and the current state of the process. Using this information, the desired action of the controller element is to compensate for changes in the process and its setpoint. This balancing act requires negative feedback, which is established by calculating the process setpoint and measured process value together to form an error variable $e(t)$. It is given by

$$e(t) = r - y_m(t) \tag{2.1}$$

where similarly to the Figure 2.1, the output is summed from process setpoint r and the measured process value y_m. The sign of the y_m denotes negative feedback, which gives the controller its ability to perform corrective actions in relation to the state of the process instead of amplifying them. The r can be set by a user or an upper level algorithm and the y_m measured from the actual system using a suitable feedback sensor. (Visioli 2006, p. 2)

2.1.1 Proportional element

Considering a traditional control approach, it is sensible to have a linear relation between the process error and controller output. Such an action is robust against disturbances in the process but does not compensate for longer term errors in the process. (Bennett 1993, p. 62) The proportional element of a PID controller establishes the described linear action and thanks to its versatility and reliability it is a common element in the variants of the PID controller. The output $u_p(t)$ of the proportional element is given by

$$u_p(t) = K_\mathrm{p} e(t) \tag{2.2}$$

where K_p is the gain of the proportional action and $e(t)$ the result of the error (Formula 2.1). The proportional gain is used to adjust the proportional element's magnitude of response to the error in the system. (Visioli 2006, pp. 3-5)

As described, the proportional element gives the PID controller an ability to react linearly to an error in the system. It increases the controller's output when the error in the system is large and respectively, decreases the controller's output when the error in the system is low, with a magnitude denoted by the proportional gain. In practice, a proportional gain value that is too low causes the controller to respond slowly to changes in the $e(t)$, and causes a large steady state error in the controlled

process. A higher value of the proportional gain decreases steady state error in the process, but on the other hand too high a value causes oscillatory behavior in the process. (Åström and Hägglund 1995, pp. 64-67)

2.1.2 Integral element

The second common element in the PID–based controllers is the integral element. Whereas systems that are controlled with proportional controllers tend to have some steady state error, the feature of the integral element is that it corrects for longer term errors, thus zeroing the steady state error. (Ogata and Yang 2002, p. 218) The integral element considers the accumulated error over consecutive cycles of the controller. The accumulated error is used to calculate the integral element's output $u_i(t)$, given by

$$u_i(t) = K_\mathrm{i} \int_0^t e(t)\, dt \tag{2.3}$$

where K_i is the gain of the integral element and $e(t)$ the result of the error (Formula 2.1). Low values of integral gain cause the error zeroing function of the integral element to act slowly and too high values cause the system to overcompensate and thus cause oscillatory behavior in the process. (Visioli 2006, p. 5)

In the domain of digital implementations for robotic systems, the realisation of the integral element may differ from its theoretical representation (Formula 2.3). Considering a case where the dynamics of the controlled system change depending on the state of the system, e.g. in robotic arms, it is beneficial to adjust the integral gain during system operation to compensate for the changing system dynamics. As opposed to the previously presented integral element calculation method (Formula 2.3), some modern systems such as the OptoDrive and Texas Instruments C2000 calculate the K_i into the accumulated integral. This allows adjustment of the integral gain during controller operation while avoiding jumps in the controller output when the gain is modified. The modified integral element's operation is given by

$$u_{imod}(t) = \int_0^t K_\mathrm{i} e(t)\, dt \tag{2.4}$$

whose value is stored into a variable that is kept in global memory. This preserves and accumulates the corrective action of the integral element on each control cycle. (*C2000™ Digital Controller Library* 2015, p. 23) This kind of an integral action allows an upper level software to set different controller gains for different system states, to have an optimal controller operation at every point of time.

2.1.3 Derivative element

If a process is controlled with the proportional and integral elements, all the compensating actions of the controller are based on errors that are already present in the process. Therefore, before the errors are compensated, they already affect the perceived performance of the controller. The derivative element of the PID controller can respond to a rate of change of the error $e(t)$, thus increasing the sensitivity of a controller and allowing an early correction for errors before they grow to be significant. (Ogata and Yang 2002, p. 222) The output of the derivative element is given by

$$u_d(t) = K_\mathrm{d} \frac{de(t)}{dt} \tag{2.5}$$

where K_d is the gain of the derivative element and $e(t)$ the result of the error (Formula 2.1).

If the derivative gain is too low, the effect of the derivative element will be lower than optimal. Too high derivative gain causes the controller to overcompensate and oscillate. High values of derivative gain also tend to apply compensative action to the process according to measurement noise, which can be heard as a hiss in some servo motor systems, for example. While a convenient element in theory, its noise-amplifying issue is a concern for real-world systems. (Visioli 2006, pp. 6, 9)

2.2 Full PID controllers

The aforementioned PID controller's elements can be combined in different ways when a complete controller is formed from them. Elements can be entirely left out of the implementation to simplify the resulting controller or to avoid problems of incompatibility, such as with the derivative controller when the process measurement is noisy. The outputs and gains of the controller elements can be made to affect each other by multiplying them together. For example, in the interacting or ideal form PID controller adding proportional gain also increases the control authority of the integral and derivative elements. The ideal PID controller is given by

$$U_i(t) = K_\mathrm{p} \left\{ e(t) + \frac{1}{T_\mathrm{i}} \int_0^t e(t)\, dt + \frac{1}{T_\mathrm{d}} \frac{de(t)}{dt} \right\} \tag{2.6}$$

where all three previously presented controller elements can be distinguished.

From the formula, it can be seen that in the ideal controller, the integral gain has been replaced by an integral time T_i and the derivative gain by the derivative time T_d. Other typical forms of the PID controller are the non-interacting or series form and the parallel form controller (see Visioli 2006, pp. 7-8 and 13-16 for further study). The parallel form is the most flexible of these, as it allows all the controller elements to be entirely turned off by manipulating their gains. It is given by

$$U_p(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t)\, d + K_\mathrm{d} \frac{de(t)}{dt} \tag{2.7}$$

where the individual controller element Formulae (2.2), (2.3) and (2.5) can be easily distinguished. (Y. Li, Ang, and Chong 2006, pp. 32-33, Åström and Hägglund 1995, p. 64)

The first PID controllers were implemented using pneumatic and mechanical assemblies. There has been a slow transition to modern digital controllers. As the modern controllers have evolved, varying implementations of the controllers' internal architecture have been used. Hence, the auto-tuning methods available for the PID controller are intended for various forms of controllers as well. Considering controller auto-tuning, the variation in the intended controller form for the auto-tuning method poses an issue for calculating the resulting tuning parameters. (O'Dwyer 2009, pp. 5-6, 19–20) The conversion between the tuning parameters for ideal, series and parallel controller forms is given by

$$K_{\mathrm{p}p} = K_{\mathrm{p}i} \tag{2.8}$$

$$K_{\mathrm{i}p} = K_{\mathrm{p}i} K_{\mathrm{i}i} \tag{2.9}$$

where $K_{\mathrm{p}p}$ is the proportional gain of a parallel form controller and $K_{\mathrm{p}i}$ is the proportional gain of an ideal form controller. For the integral conversion, $K_{\mathrm{i}p}$ is the integral gain of a parallel form controller and $K_{\mathrm{i}i}$ the integral gain of an ideal form controller. (Visioli 2006, pp. 7-8) The same formulae can be used to convert parameters between a series controller and a parallel controller, provided that the derivative element of the controllers is not used. (Skogestad 2003, p. 304)

Additionally, the gain of the integral element can be presented as integral time, which is an inverse of the integral gain. Generally, the selection of which one to use when implementing a controller depends on the application. Faster acting control loops, such as motor controllers use integral gain and slower applications, such as heaters use integral time. The conversion from the integral time to the integral gain K_i is given by

$$K_{\mathrm{i}} = \frac{1}{T_{\mathrm{i}}} \tag{2.10}$$

where T_{i} is the integral time. (Visioli 2006, pp. 7-8) It can be seen from the equation that to disable the integral element of a controller that uses integral time, the value of integral time should be set to infinity. For a controller that uses integral gain, it is sufficient to simply set the gain to zero to disable the element.

2.3 Model–based PID controller auto-tuning methods

When a new robot is designed, suitable controller tuning parameters for its servo axes are unknown. As suitable tuning parameters vary between robots, previously found parameters cannot be used. Finding the tuning parameters for the system's controller is critical for the correct operation and optimal performance of the system. A PID controller's tuning parameters can be automatically determined using system model–based and model–free methods (see Visioli 2006, p. 18, Lequin et al. 2003, p. 1023).

Model–based methods utilize an experiment to form a mathematical representation of the system, from which suitable controller tuning parameters can be calculated. In the experiments, a certain stimulus, such as a step in controller setpoint is applied on the system. The system's response to the setpoint change is then observed and measured. The resulting data is analysed to find mathematical indices that describe the system. (Leva and Maggio 2012, pp. 45-47) On the contrary, model–free methods disregard the model of a system and skip directly into modifying the controller's tuning parameters while measuring the performance of the system. These model–free methods are based on iteratively minimizing a controller performance-related fitness function, and when an optimum value of the fitness function is achieved, the tuning is finished. However, these model–free techniques require some initial guess of PID tuning parameters to begin the tuning process. The further away the initial guess is from optimal, the more iterations it takes to reach a satisfactory result. (Lequin et al. 2003, p. 1032) In the context of this thesis, the target systems for the PID controller auto-tuning will vary from small voice coil actuators to large linear axes, whose controller tuning parameters may vary significantly. Their controller tuning parameters can be completely unknown, so a model–based method is required to find a set of working controller tuning parameters, that can later be optimized.

As described, the model–based auto-tuning methods can find a mathematical representation of the system by using certain experiments denoted by the methods. A system model that is utilized by several auto-tuning methods and their model-finding experiments is the first order plus dead time (FOPDT) model. This model can

directly be used to calculate suitable controller tuning parameters (see Visioli 2006, p. 16). It is given by

$$P(s) = \frac{K_{\text{pg}}}{\tau_c s + 1} \mathrm{e}^{-\theta_d s} \tag{2.11}$$

where K_{pg} is the process gain, θ_d is the process dead time, and τ_c is the time constant of the process. The process gain is the relative change in the process induced by a change in the controller output. The dead time is the time it takes for the process to react to a change in the controller output. The time constant is the time it takes for the process to reach 63 percent of its final change when the controller output changes. These parameters can be used to model processes, such as the servo-control process of an axis' velocity. (Visioli 2006, p. 16) On the other hand, some auto-tuning methods such as the Ziegler-Nichols method calculate their own system indices to determine the PID controller's tuning parameters (see Ziegler and Nichols 1942).

The model-finding experiments used by the auto-tuning methods can be divided into two main categories, open-loop experiments with the controller bypassed and closed-loop experiments with the controller active (Pessen 1994). In both categories, setpoint response experiments are commonly used. Additionally, an ultimate gain experiment by Ziegler and Nichols has established itself as a historical tuning method for PID controllers. Instead of relying on the setpoint response of a process, it analyses the process in an oscillatory state (Leva and Maggio 2012, pp. 45-59, Ziegler and Nichols 1942). Methods that inject various stimuli to the target system exist. Some of them even utilize a stimulus that is closer to noise than actual steps or functions in the system setpoint (Gyöngy and Clarke 2006, pp. 149-162). As the model-finding experiments differ from each other, so does their actual objective behavior on a servo axis.

PID auto-tuning methods, utilizing different simple model finding experiments, were investigated for testing in this thesis. This excludes auto-tuning methods relying on the mathematical system transfer functions and frequency responses, as their computational complexity (see Åström and Hägglund 2001, pp. 1164-1168) is deemed unnecessary for the first step of the OptoDrive tuning, whose intent is to produce reliable tuning results quickly. Some methods, such as the Internal Model Control (IMC) method (see Rivera, Morari, and Skogestad 1986) were left out due to being practically superseded by the more feature complete Simple Internal Model Control (SIMC) method (Skogestad and Grimholt 2012, pp. 149-153). The chosen auto-tuning methods were picked based on their popularity and feature sets. The Ziegler-Nichols method, commonly mentioned in comparison articles for auto-tuning methods (see Tan et al. 2006, Haugen 2010), was chosen due to its popularity and historical status as the first PID auto-tuning method (Leva and Maggio 2012, pp. 45-59). Auto-tuning

methods that use the open-loop and the closed-loop setpoint response experiments were chosen to be the Cohen-Coon and the setpoint overshoot methods, also thanks to their commonness in literature (Tan et al. 2006, Haugen 2010). The last auto-tuning method, SIMC, was selected for testing due to its ability to utilize both the open-loop and the closed-loop setpoint response experiments. The mathematical formulae in the SIMC method calculate model parameters from the results of both experiments, thus allowing the use of either experiment with any tuning method that relies on the other (Skogestad and Grimholt 2012, pp. 149-153). To differentiate between the four selected methods outlined above, algorithms describing each methods' model finding experiment can be found in the sections below.

2.3.1 Ziegler-Nichols

In 1942, Ziegler and Nichols published two controller auto-tuning methods, based on two different experiments for finding the system model. One of these methods has become widely known as the Ziegler-Nichols method. It utilizes a novel ultimate gain experiment, where the controller is briefly brought into an oscillatory state. The original research paper examines and documents the different control elements of a PID controller and concludes with a proposition of the now famous tuning method. The Ziegler-Nichols method was developed on a Fulscope 100 -controller but was generalized to allow it to be utilized for other types of systems as well. (Ziegler and Nichols 1942)

Algorithm 2.1 Ziegler-Nichols ultimate gain experiment

```
1:  procedure tuneAxis(axis)
2:      setControllerGain(K_i, K_d = 0)                    ▷ Preparation
3:      while not axis.oscillating do                      ▷ Experiment
4:          setControllerGain(K_p++)
5:          setControllerSetpoint
6:          K_pu ← K_p
7:      end while
8:      P_osc ← observeProcess                             ▷ Analysis
9:      result ← calculatePID(K_pu, P_osc)
10:     return result
11: end procedure
```

Adapted from Ziegler and Nichols 1942

The basic functionality of the ultimate gain experiment proposed by Ziegler and Nichols is shown in Algorithm 2.1. The procedure begins by setting the controller into a proportional-only mode. The proportional gain of the controller is then increased until sustained oscillations are detected in the system. A small excitation step is applied into the controller's setpoint to initiate the possible oscillations after increasing the proportional gain. If oscillations are not found, the proportional gain is increased again, and the excitation repeated. A typical controller response during sustained oscillation is shown in Figure 2.2. In the beginning of the figure, the excitation step causes the controller output to jump. After the jump, the controller constantly overcompensates, causing a sustained oscillation in the system. The results of the experiment are the critical gain K_{p_u}, which is the proportional gain that was used when the oscillation began, and the period P_{osc} of the oscillation. (Ziegler and Nichols 1942)

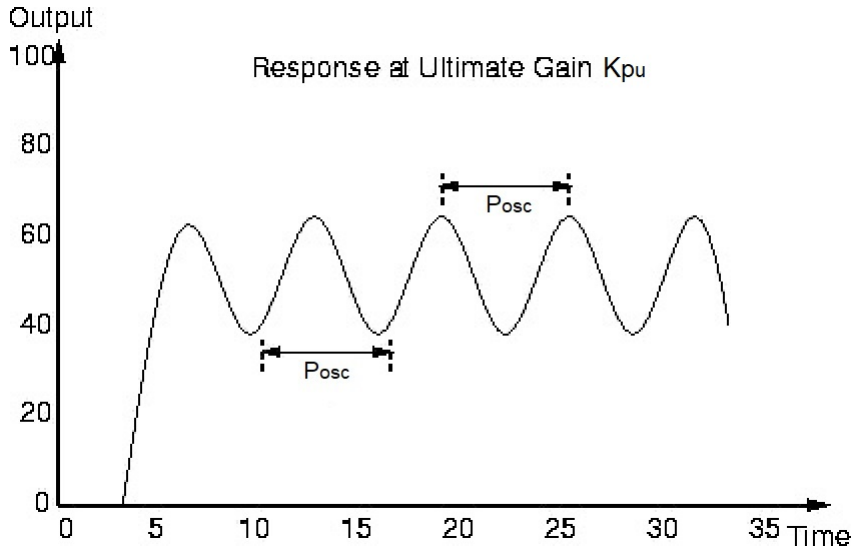

Figure 2.2 *System response in the ultimate gain experiment (adapted from Ponton 2007)*

The Ziegler-Nichols method calculates controller tuning parameters for a system based on the critical gain and the oscillation period. Multiple variants of tuning parameter calculation formulae exist for the Ziegler-Nichols method, proposed by authors that have put effort into optimizing the method. For the purposes of this thesis, three of the most commonly occurred variants were selected, namely the classic Ziegler-Nichols rules, a "some overshoot variant" and a "no overshoot" variant of the formulae. Table 2.1 summarizes formulae for calculating the resulting controller tuning parameters for each of the selected variants. The parameters are suitable for an ideal form controller. No single variant of the Ziegler-Nichols tuning method is the best or the most refined, but rather should be selected based on which of them works best for a given system. (McCormack and Godfrey 1998, p. 46, Skogestad 2003, p. 301, Pessen 1994, pp. 553-556)

Table 2.1 *Formulae for P and PI controller parameters in the Ziegler-Nichols method*

Variant	Target controller	K_p	T_i
Classic Z-N	P	$0.50K_{\mathrm{p}_u}$	–
	PI	$0.45K_{\mathrm{p}_u}$	$P_{osc}/1.2$
Some overshoot	PI	$0.33K_{\mathrm{p}_u}$	$P_{osc}/2.0$
No overshoot	PI	$0.20K_{\mathrm{p}_u}$	$P_{osc}/2.0$

2.3.2 Cohen-Coon

Another popular tuning method that was published in the mid-twentieth century is the process reaction curve –based method proposed by Cohen and Coon. It utilizes an open-loop setpoint response experiment originally proposed by Ziegler and Nichols. In the experiment, a step is applied to the output of a controller that is bypassed, and the resulting reaction in the process is observed. An optimal system response during the experiment is shown in Figure 2.3. The response has two distinct sections, first a steep incline in the process state in the beginning of the plot, then a smooth taper into a stable process state. The response is analysed to find parameters of the FOPDT process model (Formula 2.11). The Cohen-Coon method is more flexible than the Ziegler-Nichols, as it allows tuning processes with longer dead times, which increases the amount of use cases for it. (Cohen and Coon 1953, Ziegler and Nichols 1942, Silva, Datta, and Bhattacharyya 2007, pp. 7-8)

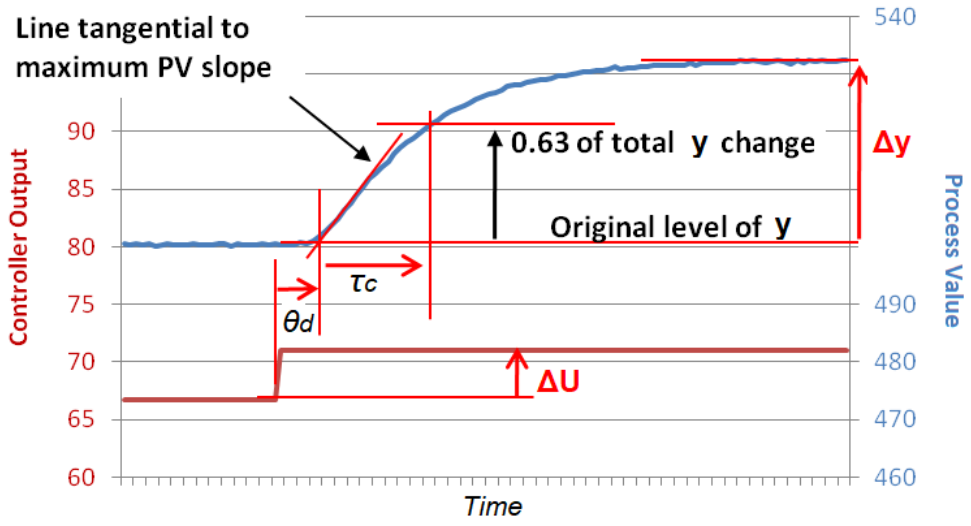

Figure 2.3 *System response in the closed-loop setpoint response experiment (adapted from Smuts 2011)*

Algorithm 2.2 Cohen-Coon open-loop setpoint response experiment

```
1: procedure TUNEAXIS(axis)
2:     DISABLECONTROLLER                                  ▷ Preparation
3:     while not stable steady state reached do           ▷ Experiment
4:         SETCONTROLLEROUTPUT(U++)
5:     end while
6:     K_pg, τ_c, θ_d ← OBSERVEPROCESS                    ▷ Analysis
7:     result ← CALCULATEPID(K_pg, τ_c, θ_d)
8:     return result
9: end procedure
```

Adapted from Cohen and Coon 1953 and Skogestad and Grimholt 2012, pp. 149-150

The operation of the open-loop setpoint experiment is presented in Algorithm 2.2. The experiment requires disabling the process's PID controller entirely and relies on manually controlling a step in the process to obtain the tuning parameters for the PID controller. From the open-loop setpoint response in the process, the process gain K_{pg}, the dead time θ_d, and the process time constant τ_c can be derived, as shown in Figure 2.3. The calculation of dead time is done by finding a tangent of the steepest slope of the process response curve. The process dead time is the time from the setpoint application to the tangent's intersection point with the time axis of the process reaction plot. The time from the intersection point to a point where 63 percent of the change in the process value has occurred is the process time constant. The process gain is given by

$$K_{pg} = \frac{\Delta y}{\Delta U} \tag{2.12}$$

where Δy is the observed relative change of the process value and ΔU the applied relative change in the controller output, both as percentages in their own ranges of operation. (Ziegler and Nichols 1942, Haugen 2010, pp. 83-84)

Table 2.2 *Formulae for PI controller parameters in the Cohen-Coon method*

K_p	T_i
$\frac{1}{K_{pg}}\left\{0.9\frac{\tau_c}{\theta_d}+0.083\right\}$	$\tau_c\left\{\frac{3.33\theta_d/\tau_c+0.31(\theta_d/\tau_c)^2}{1+2.22\theta_d/\tau_c}\right\}$

Finally, the obtained FOPDT parameters can be used to calculate suitable tuning parameters for the target controller. Originally, the Cohen-Coon method expressed

the gain of the controller's integral element as the repetition of reset actions per minute. This element has since been adapted for integral time that modern PID controllers utilize. The formulae for calculating the Cohen-Coon method's proportional and integral gains for an ideal-form PI controller are presented in Table 2.2. The units of time for the process model should match the units used in the internal structure of the target controller. (Cohen and Coon 1953, O'Dwyer 2009, pp. 26-28)

2.3.3 Setpoint overshoot

Due to the relative complexity of existing PID controller auto-tuning methods, the methods may require time-consuming testing to obtain a satisfactory tuning result. Also, bringing a process to a state of oscillation (Ziegler-Nichols method, Section 2.3.1) or running it in an open-loop (Cohen-Coon method, Section 2.3.2) may be problematic or at least undesirable for some processes. In a robotics application, an oscillatory control process shakes the robot, which may damage the robot hardware. A solution proposed by Shamsuzzoha and Skogestad, the setpoint overshoot method, relies on a simple closed-loop setpoint response experiment that allows the target controller to remain in proportional mode. It does not require the process to oscillate. The authors note that the typical proportional gain during the experiment is roughly half of the gain required by the ultimate gain experiment. This avoids oscillation of the process. The parameters required by the setpoint overshoot method have been designed to be easily readable from the closed-loop setpoint response experiment. (Shamsuzzoha and Skogestad 2010, pp. 1220-1225)

Algorithm 2.3 Setpoint overshoot closed-loop setpoint response experiment

```
1: procedure tuneAxis(axis)
2:     setControllerGain(K_i, K_d = 0)                    ▷ Preparation
3:     while not overshoot criterion reached do           ▷ Experiment
4:         setControllerGain(K_p++)
5:         setControllerSetpoint(Δy_s++)
6:     end while
7:     t_p, b, A_o ← observeProcess                       ▷ Analysis
8:     K_p0 ← getControllerGain(K_p)
9:     result ← calculatePID(K_p0, t_p, b, A_o)
10:    return result
11: end procedure
```

Adapted from Skogestad and Grimholt 2012, pp. 150-152

The overall function of the setpoint overshoot method is shown in Algorithm 2.3. The ideal behavior of the process during the setpoint response experiment is shown in Figure 2.4. In the plot, the process variable has a steep incline after setpoint change, followed by an overshoot. After the overshoot and a few oscillations, the process settles into a steady state. This system behavior can be analysed to find the parameters required for the use of the SIMC method. The parameter t_p is the time from setpoint change to the first peak in the process. The relative process steady state change is given by

$$b = \frac{\Delta y_\infty}{\Delta y_s} \tag{2.13}$$

where Δy_∞ is the process steady state change, and Δy_s is the applied setpoint change. The relative overshoot in the process is given by

$$A_o = \frac{\Delta y_p - \Delta y_\infty}{\Delta y_\infty} \tag{2.14}$$

where Δy_p is the peak process change. The in-experiment proportional gain K_{p0} of the target controller is not explicitly defined by the method. It should simply be sufficient to obtain 10–60 percent overshoot of the process value in relation to its steady state. (Shamsuzzoha and Skogestad 2010, pp. 1220-1225)

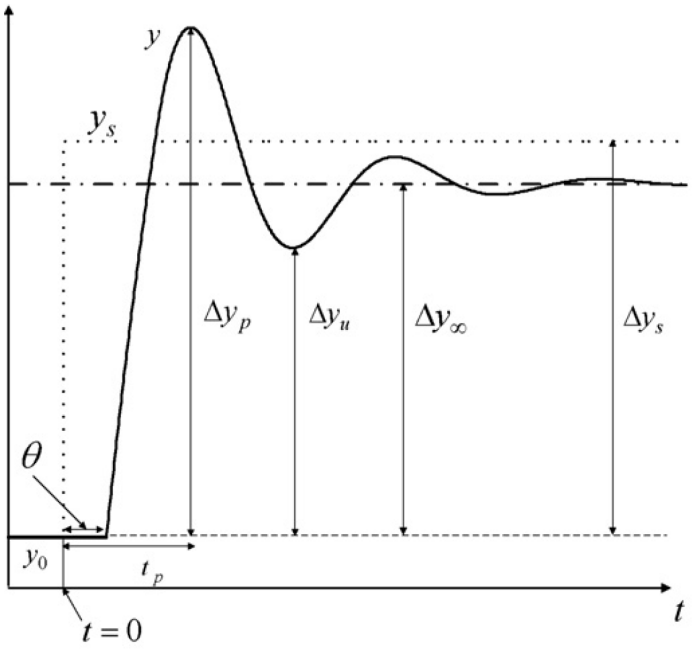

Figure 2.4 *System response in the closed-loop setpoint response experiment (Shamsuzzoha and Skogestad 2010)*

After the setpoint response experiment and its data analysis is completed, the tuning

parameters for a PI controller can be calculated. Calculations for a proportional-only controlled system were not found in literature. The final PI tuning parameters are given by

$$A = 1.152A_o^2 - 1.607A_o + 1 \tag{2.15}$$

$$K_{\mathrm{p}} = \frac{K_{\mathrm{p0}}A}{\tau_t} \tag{2.16}$$

$$T_{\mathrm{i}} = min\left\{0.86A\left|\frac{b}{1-b}\right|t_p, 2.44t_p\tau_t\right\} \tag{2.17}$$

where τ_t is an adjustment parameter, which is normally $\tau_t = 1$, but can be set $\tau_t > 1$ to "de-tune" the controller to gain a more robust tuning. Values of $\tau_t < 1$ can respectively be used to speed up and tighten the response of the controller to errors in the process. The controller tuning parameters produced by the method are intended for an ideal-form controller. (Shamsuzzoha and Skogestad 2010, pp. 1222-1225)

2.3.4 Closed-loop SIMC

The final PID parameter tuning method selected for testing in this thesis is the SIMC method proposed by Skogestad. It is an improvement on the widely spread IMC method (see Rivera, Morari, and Skogestad 1986). While the IMC method produces tuning parameters that give an adequate response to setpoint changes, SIMC is shown to improve the disturbance rejection and thus the stability of the target controller. An advantage of SIMC is that it is designed to allow an easy adjustment between operational performance and robustness of the resulting tuning parameters (Skogestad 2003, p. 291). In literature, the SIMC method has been found to provide exceptionally well performing controller tuning parameters (see Grimholt and Skogestad 2012, pp. 5-11). It was selected for testing in this thesis thanks to its apparent performance and ability to use either an open-loop or a closed-loop setpoint response experiment (see Skogestad and Grimholt 2012, pp. 149–153).

Skogestad and Grimholt propose two methods of obtaining the process model information for the use in the SIMC method. A similar open-loop setpoint response experiment as used in the Cohen-Coon method (Algorithm 2.2) was initially proposed for use with the SIMC method. Afterwards a closed-loop experiment was proposed as an alternative to increase the method's flexibility, as some systems are not compatible with the open-loop experiment. The closed-loop model-finding

experiment is identical to the model-finding experiment of the setpoint overshoot method (Algorithm 2.3), and simply extends its formulae (Skogestad 2003, pp. 292-294, Skogestad and Grimholt 2012, pp. 149-153). Additionally, it has been noted in literature that the closed-loop experiment yields superior results compared to the open-loop experiment (Hjalmarsson, Gevers, and De Bruyne 1996, pp. 1667-1669). The closed-loop experiment was regarded as more useful for the purposes of this thesis, since it would obviously keep the servo system in a more deterministic state. Therefore, the focus of this thesis regarding SIMC is on the closed-loop setpoint experiment.

The formulae for calculating the resulting tuning parameters differ in the closed-loop SIMC method, when compared to the similarly behaving setpoint overshoot method (Section 2.3.3). The formulae in the closed-loop SIMC obtain parameters for the FOPDT process model (Formula 2.11) and form a link between the parameters obtained from both open-loop and closed-loop experiments. This can be easily seen from the closed-loop SIMC method's equations. It utilizes the previously presented method of calculating overshoot amplitude (Formula 2.15) and

$$R = 2A/b \tag{2.18}$$

followed by system model-describing formulae, given by

$$K_{pg} = \frac{1}{K_{p0}b} \tag{2.19}$$

$$\theta_d = t_p \left\{0.309 + 0.209\mathrm{e}^{-0.61R}\right\} \tag{2.20}$$

$$\tau_c = R\theta_d \tag{2.21}$$

where the parameters K_{pg}, θ_d and τ_c directly correspond to the parameters used in the FOPDT model (Formula 2.11). (Skogestad and Grimholt 2012, pp. 150-153).

Finally, the resulting PID controller tuning parameters can be calculated from the FOPDT process model parameters, regardless of whether the parameters were acquired using an open-loop or a closed-loop test. The SIMC method was originally developed for a PID-type controller. The authors of the method later derived tuning rules for an ideal-form PI-controller, given by

$$K_\mathrm{p} = \frac{1}{K_{pg}} \frac{\tau_c + {}^{\theta_d}/_3}{\tau_t + \theta_d} \tag{2.22}$$

$$T_{\mathrm{i}} = min\left\{\tau_c + \frac{\theta_d}{3}, 4\left(\tau_t + \theta_d\right)\right\} \tag{2.23}$$

where τ_t is an adjustment parameter, with values ranging from $\tau_t = 0.5\theta_d$ to $\tau_t = 1.5\theta_d$. Within this range, the resulting tightness of the controller tune varies from aggressive to robust, respectively. The authors of the method note that a value of $\tau_t = \theta_d$ produces robust but somewhat conservative controller tuning parameters compared to other PID auto-tuning methods. (Skogestad and Grimholt 2012, pp. 156-164, Grimholt and Skogestad 2012, pp. 5-9)

2.3.5 Summary of the auto-tuning methods

In the previous sections, the Ziegler-Nichols (Section 2.3.1), the Cohen-Coon (Section 2.3.2), the setpoint overshoot (Section 2.3.3), and the closed-loop SIMC (Section 2.3.4) controller auto-tuning methods have been presented. The purpose of this section is to summarize the methods' key similarities and differences. All the methods consist of an experiment that is run on the target system, data analysis, and tuning parameter calculation.

The most notable differences are in the experiments themselves, as all the methods' data analysis and formulae are tied to the experiments. The auto-tuning methods utilize three different system model finding experiments. The Ziegler-Nichols, the setpoint overshoot, and the closed-loop SIMC methods allow the target controller to remain in operational proportional control mode. However, the Cohen-Coon method requires the target controller to be disabled.

The Ziegler-Nichols method is the only presented method that requires sustained oscillation in the target process. The setpoint overshoot and the closed-loop SIMC methods, utilizing the same system model finding experiment, require only momentary process overshoot. The Cohen-Coon method does not require any overshoot in the target process.

From the perspective of the method's actual use and data analysis, the Cohen-Coon method, the setpoint overshoot method, and the closed-loop SIMC method require only simple data points to be found from process reaction curves that are measured during the experiments. However, the Ziegler-Nichols method requires a period of the process oscillation to be found. The oscillation must also be sustained to be valid for the calculation of the controller tuning parameters. Based on these requirements, the analytical side of the Ziegler-Nichols method is more demanding than that of the other presented auto-tuning methods.

2.4 PID performance indices

The selection of a controller's tuning parameters is crucial for obtaining sufficient controller performance. Therefore, it is evident that a method or an index for evaluating a controller's performance is required to compare the tuning parameters produced by the tested auto-tuning methods, and to eventually answer the research questions. Various of indices for the performance evaluation of a PID controller have been suggested in literature, for example functions that integrate the controller's error $e(t)$ over time, simple indices like maximum controller overshoot, and complex mathematical formulae that evaluate a controllers' sensitivity, among others. (Vilanova and Visioli 2012, p. 82) Efficient performance measurement requires a systematic method, whose result can be calculated from sampled data and then compared together with previously calculated indices. Systematic methods that run a standardized test and gather data for calculating the performance indices appear to vary case-by-case in literature. (Haugen 2010, p. 81, Veronesi and Visioli 2010, pp. 263-268) This part of the literature review aims to find suitable performance indices for use in this thesis. These indices should be commonly accepted in the industry to provide reliable and scientifically acceptable performance results.

A PID controller's performance can be divided into matters discussing the tightness and the robustness of control. Tightness can be quantified by observing the behavior and magnitude of the error term in a PID controller over a period of time. Robustness of control focuses on the ability of the controller to perform well in situations where the noise or dynamic of the system changes over a period of time. When demanding good robustness in a system, there is a trade off in the tightness of the controller. Tightness-evaluating performance indices were found to calculate their output based on either momentary or integrated errors, the latter being more common. (Tan et al. 2006, Shinskey 1990, Vilanova and Visioli 2012, p. 142)

The discussion about controller tightness and robustness is a multi-objective optimization issue. This thesis focuses on the tightness of the controller behavior. This stems from the steep requirements on the performance of measurement robotics. Some robots made by OptoFidelity are required to reach micrometre-scale accuracy. Therefore, as integrating the controller's (tracking) error over time is a generally accepted performance measure (see Skogestad and Grimholt 2012, pp. 160-164), it is the preferred performance measurement and evaluation method for this thesis.

As described, the integrating performance indices utilize the error $e(t)$ of a PID controller to calculate the performance of the controller. Notably, the same error parameter is used in PID controllers to calculate their output. The most common performance index found in the literature review is the Integrated Absolute Error (IAE) index, which is used in various publications to measure the performance of

controllers and tuning parameters created with controller tuning methods that were being proposed. (Shinskey 1990, p. 1, Skogestad 2003, p. 297, Veronesi and Visioli 2010, p. 262, Visioli 1999, p. 589) Furthermore, Integrated Squared Error (ISE) and Integrated Time-weighted Absolute Error (ITAE) are noted to be among the most used indices (Leva and Maggio 2012, p. 52). More rarely found combinations of error and time, such as the Integrated Time-weighted Squared Error (ITSE) and the Integrated Squared Time and Error (ISTE) are left out of the presented formulae for simplicity (Tan et al. 2006, p. 1418). However, for cases that explicitly demand different focus on the magnitude or time of appearance of the error, these more rarely found indices are likely worth looking into.

The formulae for the IAE, ISE and ITAE performance indices are given by

$$\text{IAE} = \int_0^\infty |e(t)| \, \mathrm{d}t. \tag{2.24}$$

$$\text{ISE} = \int_0^\infty e(t)^2 \, \mathrm{d}t. \tag{2.25}$$

$$\text{ITAE} = \int_0^\infty t|e(t)| \, \mathrm{d}t. \tag{2.26}$$

where t is the time when the error $e(t)$ appears in the sampled data. The presented indices are suitable for both online and offline performance analysis. In addition to performance evaluation, these indices have been used to develop theoretically optimal PID controller parameter tuning methods. (Ho, Lim, and Xu 1998, p. 1009)

All of the aforementioned indices calculate their output based on the error of the controller over a period of time. However, their weighting of the magnitude and the time of appearance of the error vary. The IAE can be regarded as the most neutral of the indices. It only takes an absolute of the error and integrates it over time. ISE takes a square of the error, thus emphasizing larger errors more. ITAE factors in time by multiplying the error at each data point with the time when it appeared, thus emphasizing errors that appear later in the process. Due to the different weighting of the errors, the controller performance results will vary based on the selected index.

3. TEST SYSTEM DESCRIPTION

The system that is used to test the functionality and performance of the controller auto-tuning methods is presented in this chapter. The desire is to enable tuning the servo axes using their own sensors (i.e. system feedback device). This allows straightforward servo drive tuning in a factory environment, where separate calibration toolkits might not be available due to access restrictions. Servo axes in OptoFidelity's measurement robots are typically either linear servo driven or rotary servo and ball-screw driven. A linear servo axis was selected as the test system while planning this thesis. This is because linear axes have a linear feedback device, that directly measures the movement of the axis' linear carriage. Rotary servo axes typically have their feedback device on the back of a rotating servo motor instead of on the axis. Having the feedback device on the axis is useful for measuring the real-world behavior of the axis, as it has zero backlash. Linear servo axes are also the preferred axis type for new robots designed by OptoFidelity.

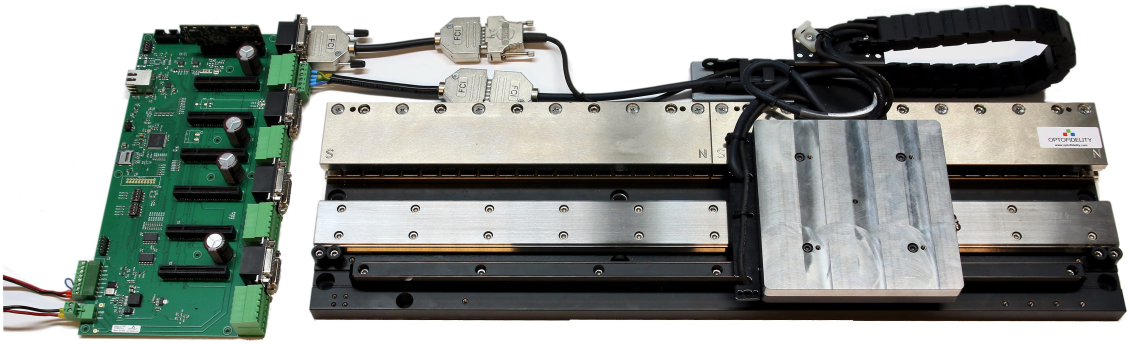

Figure 3.1 *An overview of the test system*

The feedback devices in OptoFidelity's robots are quadrature encoders with resolutions ranging from 0.1 to 10 µm. A high feedback device resolution was determined to be beneficial for the accuracy and repeatability of the measurements undertaken in this thesis. Therefore, a linear axis with a 0.1 µm resolution linear encoder was selected for the test system. The linear axis, shown in Figure 3.1, represents the typical size scale for a general-purpose measurement robot's axis. It is a self-contained unit, built on an aluminum frame. In the figure, on the left there is an OptoDrive

setup. It consists of an OptoDrive card and an OptoMotion motherboard, that are configured to drive the linear axis. The system is powered from a standard 24 V laboratory power supply. The manufacturers and model numbers of the system's parts are summarized in Appendix A.

3.1 Robot mechanics

The selected linear axis unit is designed by OptoFidelity. It utilizes a frameless linear motor system that is intended to be integrated into custom hardware. Frameless motors consist of a separate magnet track and electromechanical forcer (coil) units. These motors require a separate mechanical system with bearings and a feedback device to operate. When the forcer of the motor is energized, it produces a magnetic force against the magnet track, which is proportional to the electrical current used (see H. Li 2002). The linear axis is built on an aluminum base plate, which provides mounting for a linear bearing guide rail, a magnet track of the linear motor and an optical scale of the feedback device. A linear carriage is mounted on bearing blocks, that slide on the linear bearing guide rail. The forcer of the linear motor system and an optical scale reader unit (read head) are mounted on the linear carriage. The axis is shown partially disassembled, with the aforementioned components visible in Figure 3.2.

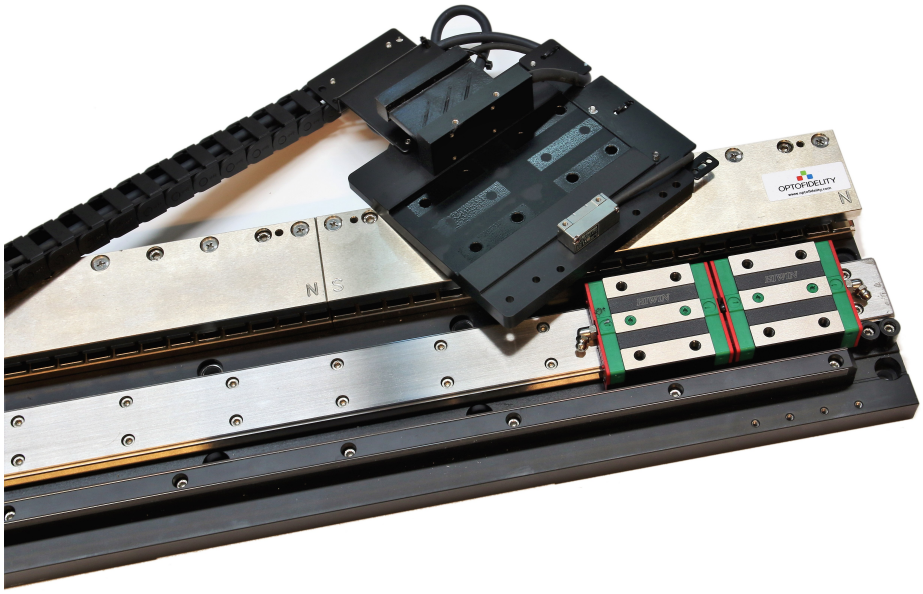

Figure 3.2 *Test axis partially disassembled*

Thanks to its high-resolution feedback device and solid construction, the selected axis is very suitable for the purposes of this thesis. However, the hardware by itself

only represents one system. Many different robots with different masses are being used by OptoFidelity. To manipulate the mass of the test system and mimic other kinds of robot axes with different moments of inertia, two mass plates were designed and manufactured from SS304 steel. They have mounting patterns to allow attaching them rigidly to the carriage of the linear axis. The mass plates, shown in Figure 3.3, weigh 1.8 kg and 3.2 kg. They are referenced in this thesis as the light mass plate, and the heavy mass plate. They were designed to be stackable if even heavier mass is desired, but during testing it was found that their combined mass of 5 kg is too heavy for the test axis and causes the motor's power limit to be exceeded. Consequently, the test system is usable with three mass configurations: with no additional mass plate, with the light mass plate, or with the heavy mass plate.

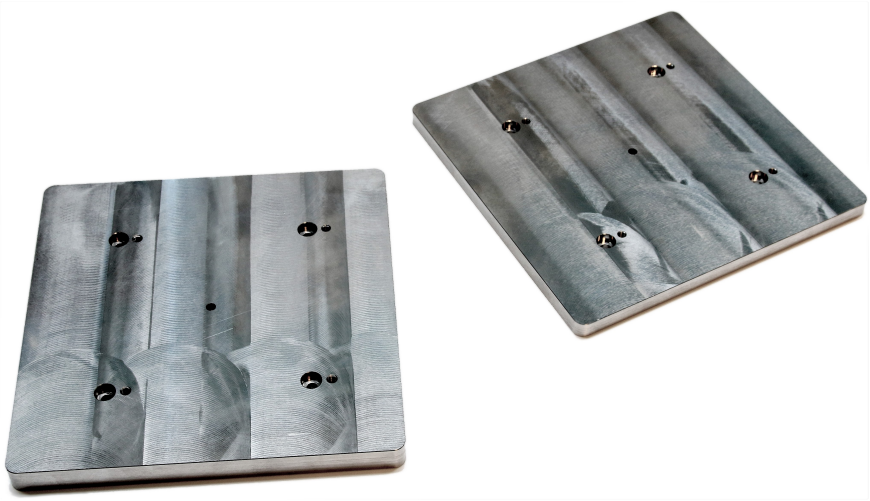

Figure 3.3 *Accessory mass plates for the linear axis*

3.2 Feedback and controller

As described, the feedback device mounted on the linear axis will be used for both servo control and measurement of the axis' behavior. The feedback device on the selected axis is based on optical measurement and consists of a read head and a scale stripe. The read head of the system tracks lines that are inscribed on the optical scale and produces a signal every 0.1 µm for the OptoDrive servo drive. Interpolation between the inscribed lines is used to provide the high-resolution measurement, with some measurement uncertainty as a trade off despite filtering (see Cheung 1999).

The OptoDrive, shown in Figure 3.4, uses a P controller to control a servo motor's position and a PI controller to control its velocity. Both are derived from an ideal form controller (Formula 2.6). The integral element of the OptoDrive's velocity controller uses integral gain instead of integral time as its control parameter. The controllers are organized in a cascaded form, where the position controller's output

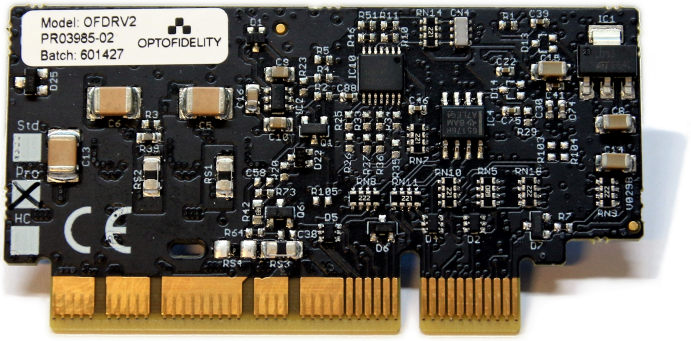

Figure 3.4 *The OptoDrive servo drive*

is the velocity controller's setpoint. The velocity controller's output is fed into the power stage of the controller to produce a force in the controlled servo motor. Feedback for the position controller is directly read from the feedback device of the system, while the feedback for the velocity controller is derived from the position data. Differentiating the step-shaped position data causes noise in the velocity information. Because of the noise in the data, the inherently noise-sensitive derivative element of the velocity controller is not used. Additionally, the position controller of the system is equipped with a feed-forward function, which calculates an ideal position controller output (velocity setpoint) to track the changing position setpoint during trajectories. This allows the position controller to be realised with a simple P controller.

Special consideration was put into the method that reads the required data for the performance measurement and controller tuning from the OptoDrive. The data reading method should be robust and easily usable. The OptoDrive allows sampling a comprehensive set of system parameters into a 2048 -slot buffer with a sampling rate of up to 2500 Hz. The sampled parameters can be selected from a list of main metrics for a servo system, including the position and velocity controllers' setpoints and outputs. These values describe the current and desired status of the system and can therefore be used to calculate the error parameters of the controllers. The drive-side sampling allows all the required data to be gathered, but no readily available method existed for downloading the sampled data for convenient use in a PC application. A C++ -application was created to read the measured data from the drive's buffer over a TCP/IP connection, and to store it into a comma-separated values (CSV) file. The application allows its user to select sampled parameters, select the sampling rate, and read the data from the drive into a CSV file. This allows the data from the axis to be available in any system that utilizes the OptoDrive for servo control, without additional hardware requirements. It appears that there was a demand for such an application, as it has already been taken into wide use in OptoFidelity.

4. STANDARDIZED TESTS

Testing the PID controller auto-tuning methods in this thesis consists of observing and analysing the general behavior of the methods, considering their suitability for tuning measurement robots, and ultimately measuring the real-world performance of the parameters produced by them. This requires a standardized controller performance measurement test. In addition to testing auto-tuning methods, the performance measurement test can be used to test quantify the tuning performance of humans as well. Outside the scope of this thesis, the performance measurement test can be used as a robot quality control procedure. In this chapter the controller performance test is presented first. A PID controller auto-tuning method test, and a quantitative interview where the target axis of this thesis is tuned by humans are defined afterwards, both relying on the controller performance measurement test.

4.1 Controller performance measurement test

As described in the Section 2.4 of the literature review, PID controller performance can be calculated using indices, that base their operation on integrating the error of the controller during a standardized test. The standardized test should be defined on a case-by-case basis for a specific system. For a measurement robot, it is logical to define the test based on the robots' real-world operation. Consequently, a set of back and forth point-to-point positioning moves was selected as the test, during which data would be gathered, and later used to calculate the performance indices. Point-to-point moves utilize the velocity and position controllers of the OptoDrive, so performance indices for both controllers can be calculated based on the test. This requires recording the actual and desired velocity and position, a total of four parameters during the test. The data is recorded using the C++ implementation described in Section 3.2, and sampled with the highest available sample rate in order to reduce errors caused by aliasing of data.

General types of robot axes and their capabilities were considered when defining the trajectory of the point-to-point moves. After discussions with OptoFidelity's test engineers, it was concluded that a velocity of $50\,{}^{\mathrm{mm}}/_{\mathrm{s}}$ and an acceleration of $1000\,{}^{\mathrm{mm}}/_{\mathrm{s}^2}$ can be reached with a clear majority of the robots used by OptoFidelity.

It is beneficial for the test to specify the highest velocity and acceleration possible, as they are more likely to produce measurable errors in the controllers' operation. Trajectories with the specified velocity and acceleration were tested on the linear axis selected for this thesis. It was found that the longest point-to-point distance that can be done with the specified velocity and acceleration is roughly 6 mm, until the controller's measurement buffer is filled with data. The smallest axis in OptoFidelity has a total movement range of 5 mm, so that was defined as the length of the test trajectory. The velocity and position profiles of the resulting trajectory are shown in Figures 4.1 and 4.2. It utilizes almost the entire length of the OptoDrive's measurement buffer, and is usable with every servo axis currently in use in OptoFidelity. The presented trajectory moves the axis in the positive axis direction.

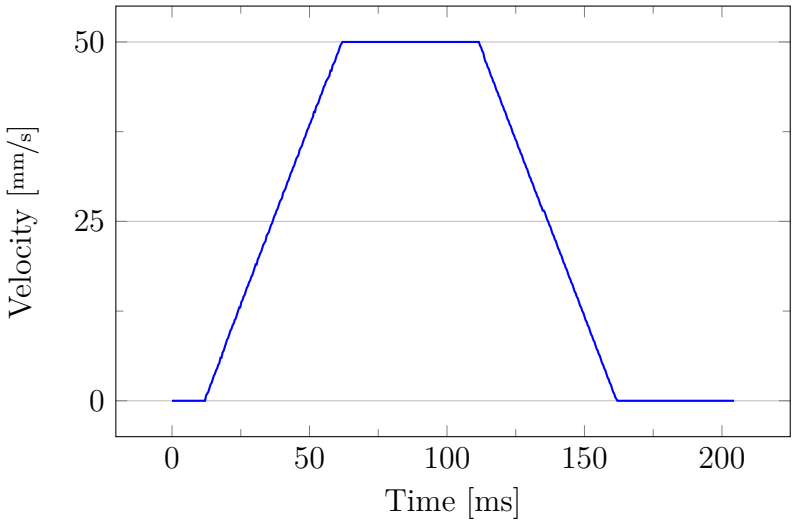

Figure 4.1 *Velocity profile of the performance test*

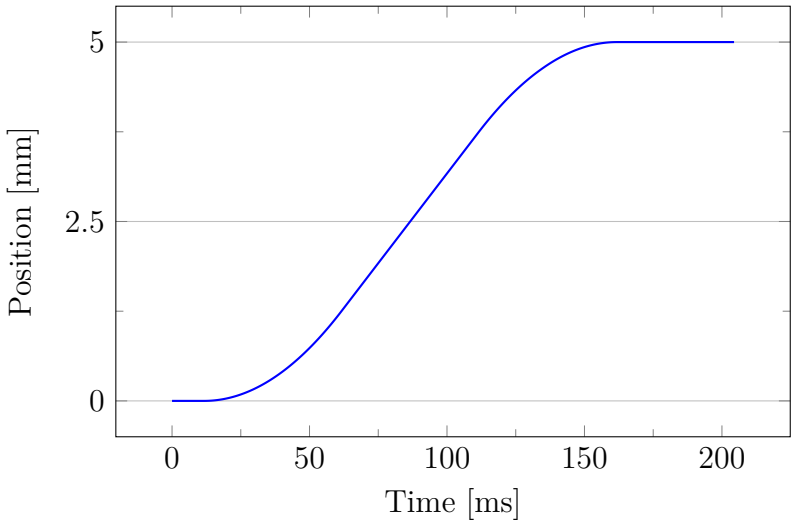

Figure 4.2 *Position profile of the performance test*

The actual test set of point-to-point moves are constructed by utilizing the presented

trajectory profile. The amount of moves to be done was selected to be five. It produces a satisfactory trade-off between the amount of recorded data and the time it takes to run the test. The moves are run in both axis directions to record possible backlash in the axis, and to save time. The negative direction moves utilize the same trajectory profiles (Figures 4.1 and 4.2) that are used in the positive direction moves, but with negative coordinates. In hindsight, six of the point-to-point moves could have been run to allow the axis to return to the start position in the end of the test with little time cost. However, when this was realised, most of the test data in this thesis had already been gathered.

The point-to-point moves were executed on the test system of this thesis to obtain data for further analysis. The heavy mass plate was installed on the axis during the test. Incorrect controller tuning parameters, given in Table 4.1, were used to emphasize controller errors. The controllers' tracking errors (see Cervantes and Alvarez-Ramirez 2001) were calculated by subtracting each measured process value from the controllers' setpoints (Formula 2.1).

The tracking errors of both controllers during all five test moves are shown in Figures 4.3 and 4.4. The positive direction trajectories are shown in red, violet, and orange, and the negative direction trajectories are shown in blue and cyan. The prominent observation from the figures is that the velocity controller performs sub-optimally especially in the beginning of the trajectories, causing a large position tracking error. This error is later corrected, until further errors in the system affect the position tracking again. The positive direction moves show more tracking error than the negative direction moves. The noise caused by differentiating the data from the feedback device can easily be seen in the velocity tracking error plots. Evidently, the points of the test trajectory where the acceleration of the system changes are the most problematic for the operation of both controllers. Based on this initial testing, there is correlation between the velocity controller's setpoint tracking performance and the perceived performance of the position controller. The tuning parameters for the velocity controller should therefore be carefully selected when testing the performance of the position controller. Both controllers' behavior is similar over the five test runs, but there seems to be less tracking error when moving to the negative direction. This can be caused by varying friction in the axis or tension on the axis carriage posed by its energy transfer chain or cabling.

Table 4.1 *Controller tuning parameters used during the tracking error testing*

	Position K_p	Velocity K_p	Velocity K_i
Gain	100	100	100

The performance measurement indices for use with the position and velocity con-

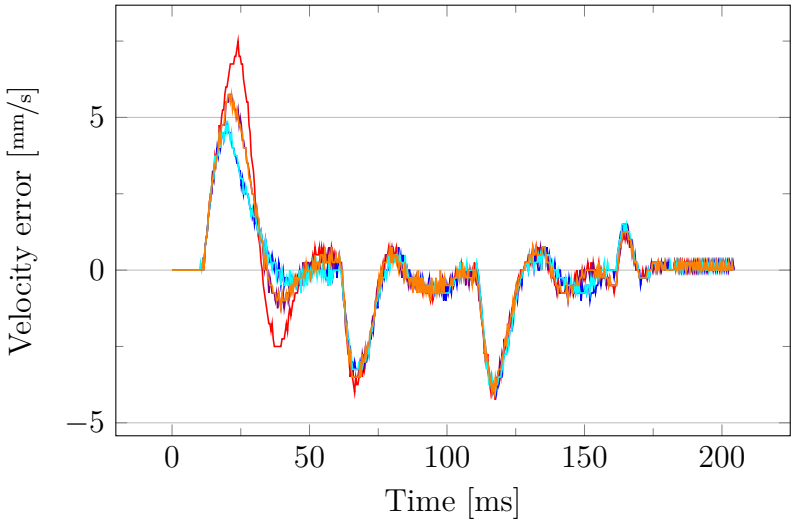

Figure 4.3 *Velocity tracking errors of performance test trajectories*

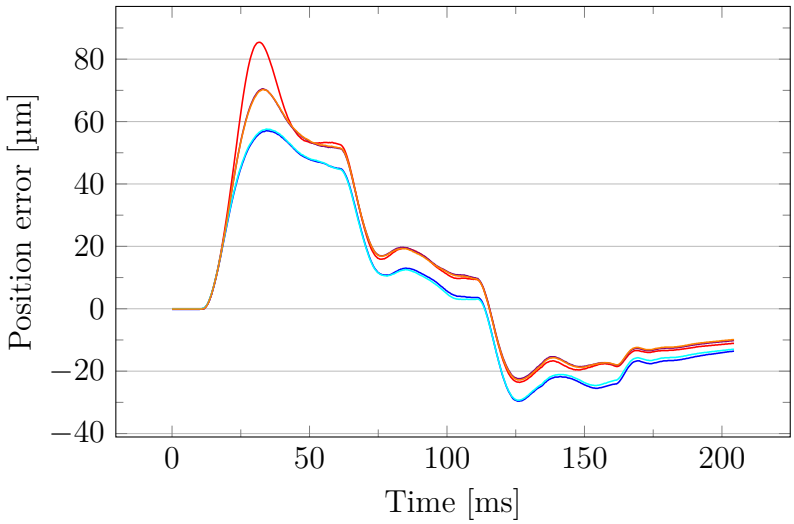

Figure 4.4 *Position tracking errors of performance test trajectories*

trollers were selected from the set of indices described in Section 2.4. Actual use cases of OptoFidelity's measurement robots were considered when selecting which indices to use. As described, the indices put different emphasis on the size and time of occurrence of the controller's tracking error. Larger errors are more critical for the robots' use cases, and thus the squared errors ISE and ITSE are the logical choice. Smaller errors, such as the encoder measurement and differentiation –induced noise to the velocity data are weighted less when squared errors are used. This is advantageous considering the noise of the test system's velocity data, seen in Figure 4.3.

The velocity controller's tracking error is critical in use cases that utilize swiping gestures, such as touch display calibration and lag testing. In these tests, the robot moves its tool on the surface of a touch display along a constant velocity vector. The

stability of the robot's velocity during swiping, and therefore momentary errors in velocity, are more interesting than the time when the errors occur. This methodology was used to select the ISE index (Formula 2.25) for the performance evaluation of velocity controller.

The time-weighting ITSE index was initially considered for the evaluation of the position controller's performance. This was because the repeatability and accuracy of the robots' positioning at the end of a trajectory is important. They are both dependent on the position controller's ability to reach its setpoint at the end of the trajectory. However, minimizing the robot's cross-track error is another important subject to consider when tuning a controller. The cross-track error can be observed when two or more axes working synchronously are attempting to follow a trajectory. Tracking errors in individual controllers can cause the final trajectory of the robot to be deformed. This is because there is no supervisory control over the error (for further study on the subject, see Shieh, A.-C. Lee, and Chen 1996). To minimize cross-track error, the time-ignorant ISE index was selected for the position controller as well. An additional scientific benefit to utilizing the ISE index for the evaluation of both controllers is that this is the most widely featured PID controller performance evaluation index found in the literature research (see Section 2.4).

Table 4.2 *ISE values of performance test point-to-point moves*

Move #	Direction	Velocity ISE	Position ISE
1	Positive	1552.5	0.4564
2	Negative	1097.1	0.3840
3	Positive	1185.7	0.3976
4	Negative	1058.7	0.3768
5	Positive	1152.4	0.3843
Mean		1209.3	0.3998

After running the specified five trajectories on the axis and calculating the tracking errors during the trajectories, ISE indices can be calculated for each of the trajectories. The tracking error data sets contain uncertainties, including measurement noise and variance in the actual controller behavior between the point-to-point moves (see Figures 4.3 and 4.4). The ISE values for the given tracking error data, and their mean values are given in Table 4.2. As described, the test data set was gathered with purposefully incorrect controller tuning parameters, so the results represent a badly tuned controller. The ISE values for the negative direction moves are smaller, which is a result of the smaller tracking errors in those moves (see blue and cyan curves in Figures 4.3 and 4.4). The first point-to-point move has a higher ISE values than the others, which is expected considering the tracking error plots for it (red curve in Figures 4.3 and 4.4). This behavior can be caused by static friction on the axis. At

the end of the test, the ISE results are combined to a mean value for the velocity and the position controllers to reduce the measurement uncertainty and variance. The resulting mean values represent the observed performance of the given system and controller.

4.2 Auto-tuning method performance measurement test

Testing the auto-tuning methods' performance relies on the performance results that are obtained using the method described in the previous section. The test axis is auto-tuned with different masses to obtain comprehensive test coverage. The mass on the axis will be varied by utilizing the mass plates described in Section 3.1. To analyse the distribution and variance for each of the methods' performance, the axis will be auto-tuned ten times with each of the three available axis mass configurations. After each auto-tuning operation, the axis performance will be tested with the newly found tuning parameters. This results in 30 different tuning parameter sets and ISE performance numbers for the auto-tuning method.

The resulting performance numbers are then analysed to allow comparison of the auto-tuning methods. The results are treated as a single data set regardless of mass configuration. The performance results for the different mass configurations are mutually comparable, as they are calculated from tracking error results obtained with a tuned controller. Therefore, the results represent how well the auto-tuning method has managed to adapt the controller to the different mass systems. For further testing on the matter and justification for combining the performance results for different masses, see Appendix D.

The average performance and repeatability of the tested auto-tuning methods are interesting, when considering the research questions. Therefore, a mean value is calculated from all the performance test results for a given auto-tuning method. A standard deviation, denoted by σ, of the performance results is calculated as well to gauge the repeatability of the method being tested. Additionally, a box plot will be created to visualize the distribution of the performance results.

In practice, a Python software platform was set up to handle the test sequence, gather the data from the axis, and calculate the controller performance. It utilizes the C++ servo drive data handling application described in Section 3.2 to transfer data from the servo drive to local variables in the Python platform. The data is transferred via a CSV file. The software platform was built to be modular and easily expandable. This was to form a foundation for implementing the controller auto-tuning methods and their model-finding experiments.

4.3 Manual PID parameter tuning test

Currently, the servo axes in OptoFidelity's robot systems are tuned manually by humans. The PID auto-tuning methods tested in this thesis are competing against manual tuning. Therefore, the performance of the auto-tuning methods should be comparable to that of a human to be considered for use. It is sensible to compare the auto-tuning methods and human tuning by using the controller performance results and not the actual tuning parameters, as the magnitudes of the proportional and integral gains can vary, and their effect on goodness of the controller behavior is not straightforward to determine. To see whether the auto-tuning methods are a feasible option for tuning the OptoDrive, a mean value of humans' tuning performance, and its standard deviation was chosen to be the baseline for comparison. A quantitative interview was planned to find these results.

In the interview, OptoFidelity personnel with experience on tuning servo drives are invited to manually tune the test axis with all three weight configurations. The planned number of human testers is five ($N = 5$). Each tester will produce three sets of controller tuning parameters for the velocity and the position controllers of the test system. The performance of the tuning parameters will then be gauged with the presented performance measurement method (Section 4.1). In addition to finding the tuning parameters and their performance, the manual tuning process will be observed to evaluate the difficulty level of the tuning process.

After finding the performance results for each axis mass configuration, controller, and tester, a total of fifteen results for both controllers of the OptoDrive, a mean value and a standard deviation are calculated for both controllers' results. The performance results for different masses can be combined like this, similarly as in the auto-tuning method performance test (for further information on the matter, see Section 4.2 and Appendix D). This results in mean and standard deviation of performance values for the velocity and the position controller of the OptoDrive. The values represent human performance on tuning the test system of this thesis.

In practice, the testers will be given a brief overview of the test axis and the additional mass plates. They are then asked to find what they regard as the optimal position controller's K_p and velocity controller's K_p and K_i parameters for the test axis. The parameters will be tuned using a standard tool that is used in OptoFidelity to configure the OptoDrive. The tool allows its user to set the controller tuning parameters to the OptoDrive. The tool also displays a real time plot of the controller's tracking error over a repeating user-defined point-to-point move. The testing plan that will be used to support the observational test is shown in Appendix B. All the data from the test will be gathered anonymously.

5. AUTO-TUNING EXPERIMENT IMPLEMENTATIONS

In this chapter, implementations for the system model-finding experiments and the auto-tuning methods are presented. Insight is given into the technical choices made during the implementation work. The controller tuning parameters can be easily calculated from the results of the model-finding experiments. Therefore, regarding the technical implementation, it is logical to focus on the model-finding experiments of the auto-tuning methods. The model finding experiments were implemented based on the research literature of the controller auto-tuning methods in Section 2.3. The python software platform that was established for the auto-tuning method testing in Section 4.2 was used as a basis for the auto-tuning method implementations. This was advantageous, as the platform was already capable of sending instructions to the test axis and gathering data from it.

Based on the literature review in Section 2.3, there are three system model-finding experiments that should be implemented for the system: the open-loop setpoint response experiment (Algorithm 2.2), the closed-loop setpoint response experiment (Algorithm 2.3) and the ultimate gain experiment (Algorithm 2.1). The experiments have common functions from the perspective of implementing them in software. The servo drive should first be prepared parameter-wise for each of the model-finding experiments. All the model-finding experiments then utilize a structure where a certain parameter of the servo drive is ramped up in steps. The behavior of the system is then observed during or after each of the steps until suitable behavior is found. After the experiments, the controller tuning parameters are calculated and applied to the servo drive. These common parts were therefore implemented first, and afterwards the implementations for the experiments written. The auto-tuning methods were then constructed from the experiments simply by adding the tuning parameter calculation formulae on top of their respective model-finding methods. The implementations were built in a generic form. This allows their target controllers to be either the velocity or the position controller.

According to the auto-tuning methods' descriptions in Section 2.3, all the methods produce tuning parameters that are compatible with an ideal-form controller. They also produce the control variable of the controller's integral element as integral time.

The tuning parameters produced by the methods are therefore compatible with the controller form of the OptoDrive, but the integral time parameters must be converted to the integral gain notation used in the OptoDrive. This conversion is done using the Formula (2.10).

5.1 Common functions

Software functions were created in Python to set up the servo drive for the model-identifying experiments. Depending on the experiment, different configurations for the OptoDrive are required. For example, the target controller should operate in proportional mode for the ultimate gain experiment and the closed-loop setpoint response experiment. However, it should be disabled and bypassed for the open-loop setpoint response experiment. A parameterized function was created to set any parameter of the servo drive. Parameter sets for each of the experiments were created, so that they could be applied to the servo drive with a single function call.

As described, all the model-identifying experiments in this thesis have some parameter, either the controller setpoint or the output of a bypassed controller, that should be gradually increased until suitable system behavior is detected. A software structure was created to allow this, with an emphasis on flexible data analysis to create a basis for calculating the key parameters of the model-finding experiments. The data analysis for the model-identifying experiments could then be implemented in separate, easily maintained functions. A function to save the resulting tuning parameters to the servo drive's non-volatile memory was also implemented. These components complete the bulk of the software platform. It only requires the data analysis for the model-identifying experiments and finally the parameterized use of the aforementioned functions.

5.2 Ultimate gain experiment

The ultimate gain experiment, outlined in Algorithm 2.1, is used in the Ziegler-Nichols auto-tuning method. The goal of the experiment is to find the critical proportional gain of the target controller, where the controlled system begins to oscillate (see Section 2.3.1). For the experiment, the controller's integral element is disabled, but otherwise the controller can be kept in normal operation. The K_p is gradually ramped up using the software structure implemented in the previous section, while observing the measured data from the system. The position controller is sent small 1 mm steps in its setpoint to initiate possible oscillatory behavior. Thanks to the OptoDrive's controller structure, these steps also affect the velocity controller to

initialize oscillation when tuning it. When a sustained oscillation occurs, the current K_p is regarded as the ultimate gain $K_{\mathrm{p}u}$.

To find whether the servo axis oscillates, a Fast-Fourier Transform (FFT) is calculated and the maximum peak amplitude in the resulting FFT plot is observed. The Fourier Transform is calculated in practice with "numpy.fft.fft" function, available in Python's NumPy library (see Scipy 2017). The controller's setpoint and process measurement data is split in half. Their individual frequency peak amplitudes are then analysed to find whether the axis' oscillation is sustained or convergent. The FFT result is also used to obtain the system's oscillation frequency, which is then used to calculate the oscillation period.

Finally, the oscillation period and the critical gain are used to calculate new tuning parameters for the controller, using formulae from the Ziegler-Nichols auto-tuning method. Multiple variants of formulae for calculating the tuning parameters were found in the literature review. The oscillation period and the critical gain are used to solve all the formulae, according to Table 2.1. All of the formulae were implemented in the software to allow further testing of them.

5.3 Open-loop setpoint response experiment

The open-loop setpoint response experiment, outlined in Algorithm 2.2, is used in the Cohen-Coon method. In the experiment, the target controller is disabled, and the process is controlled directly, hence the name open-loop. The experiment induces a step into the process control variable and observes the resulting behavior of the process. Ideally, the resulting process behavior resembles the example curve given in Figure 2.3. From the observations, key parameters of a FOPDT system model (Formula 2.11) can be identified and calculated (see Section 2.3.2).

For the practical implementation of the open-loop setpoint response experiment, the existing program structure for increasing the control variable was utilized to increase the servo motor torque to find a system model for the velocity controller. In normal operation, the servo motor torque is the output of the velocity controller. Setting it directly allows the velocity controller to be effectively bypassed. The torque is increased gradually until a stable region in the process reaction is obtained. When applying similar methodology to the position controller, it is evident that the velocity of the servo motor should be controlled to bypass the position controller. However, a process reaction curve similar to the example Figure 2.3, where the system's position would settle with some constant non-zero velocity value is illogical. Therefore, it was concluded that the open-loop setpoint response experiment is unsuitable for tuning the position controller of the OptoDrive. It is also noted in literature that systems

with no steady state should be tuned using closed-loop procedures, as finding the open-loop process gain for such a process is problematic (see Ponton 2007).

When sufficient axis torque, with stable axis velocity at the end of the open-loop setpoint response experiment has been found, the resulting process behavior can be analysed. The process behavior plot is used to find the FOPDT time constant, dead time, and process gain parameters. The actual analysis is straightforward, following the outline described in Section 2.3.2. The steepest slope is found by differentiating the process reaction curve and by finding the maximum value of the resulting plot. A mathematical tangent for the steepest slope is calculated with a simple point-slope equation of a line. The equation of the tangent can be used to find its intersection point with the time axis. The process gain can be calculated from the in-experiment torque and velocity. Both of the values are converted to percentages of their maximum values. By dividing the relative velocity with the relative torque, the process gain is found. Finally, with these parameters, the ideal-form controller tuning parameters can be calculated using the formulae presented in Table 2.2.

Finding the maximum values of the axis torque and velocity required in the method involves some work. The maximum torque value can be requested from the servo drive, which has the maximum electrical current characteristic of the servo motor saved in its memory. The maximum velocity of the axis on the other hand, can be calculated from the recorded servo output voltage during the stable velocity region of the process reaction curve. Here, a linear dependency between the axis velocity and the required servo output voltage to overcome the motor's back electromotive force (EMF) is utilized. This assumes that there is a no-external load situation, as the back EMF is the limitation of the ultimate motor velocity (see Pillay and Krishnan 1991, pp. 988-991). In the test, the axis is located on a level surface and no external forces are applied to it, so this methodology can be applied.

5.4 Closed-loop setpoint response experiment

The final experiment to implement in this thesis is the closed-loop setpoint response experiment outlined in Algorithm 2.3. It is used to identify the system model for the setpoint overshoot and the closed-loop SIMC methods. In the experiment, the controller is kept operational but with only the proportional element in use, similarly to the ultimate gain experiment. Controller setpoint steps, gradually increasing in magnitude, are sent to the controller using the existing software structure. Data from the system is recorded and analysed after applying the setpoint steps, with the goal to have a sufficient overshoot in the system response before the system settles to a stable state. The desired form of the system response is shown in Figure 2.4. The

criterion for the sufficient overshoot is 10 percent over the steady state (see Section 2.3.3).

The data is then analysed to find the key parameters t_p, b, and A_0, as described in Section 2.3.3. The authors of the setpoint response experiment intended to make the data analysis of the method as easy as possible (see Shamsuzzoha and Skogestad 2010, pp. 1220-1225). This was found to be the case, as simple program loops were sufficient for finding the key parameters. Depending on whether the setpoint overshoot or the closed-loop SIMC method is being tested, different formulae are used to calculate the resulting controller tuning values from the calculated key parameters. The tuning parameter calculation formulae for the setpoint overshoot method are given in Section 2.3.3, and the formulae for the closed-loop SIMC method in Section 2.3.4.

6. EXPERIMENTS

This chapter presents the results of the experiments conducted in this thesis. First, an evaluation of the performance of humans when tuning a PID controller is conducted. From the results, a baseline mean value and standard deviation of human tuning performance is established for both controllers of the OptoDrive.

For the four auto-tuning methods in this thesis, three system model-finding experiments were implemented. The behavior of the experiments is first analysed to find whether they are compatible with the OptoDrive, and if they are usable in a measurement robot. If deemed suitable the controller auto-tuning methods enabled by the experiments are tested for performance and compared against the human performance baseline. These tests are conducted for both controllers of the OptoDrive. The standardized tests described in Chapter 4 are used in this chapter to assess the performance of humans and auto-tuning methods. The standardized tests include repetitions to reduce the uncertainty of the results.

6.1 Human tuning performance

As a first experiment in this thesis, human performance on tuning PID controllers was tested according to the description in Section 4.3. The five test subjects saw the situation as a good opportunity to refine their skills in tuning the controller's parameters. It took roughly 20 minutes for the test subjects to create a single set of tuning parameters on the test axis, so tuning the axis with all three axis mass configurations took about an hour on average. The main purpose of this section is to analyse the produced tuning parameters' performance. Mean and standard deviation of human tuning performance will be calculated for the velocity and the position controllers. These numbers can be later used as a baseline of comparison when evaluating the auto-tuning methods.

Some general observations were made based on the discussions with the test subjects. Most importantly, tuning the servo axes was not well understood from the theory point of view. The subjects were merely optimizing the tracking errors of the controllers by the method of trial and error, similarly to how a model–free auto-tuning method would work. Some sense of the tuning parameters' effects to the

system behavior existed, but the tuning process was not systematic. All the test subjects were able to get the target servo axis working reliably and smoothly, without oscillation or other undesirable behavior in the system. Some of the test subjects commented on the difficult and specialized nature of controller parameter tuning. The following quote from a test subject sums up manual tuning well. It has been freely translated from Finnish to English.

> "There are no processes for this. Or actually there is, but who has time to learn them."

The controller tuning parameters produced by the testers are gathered in Appendix C. Based on the values of the tuning parameters, the repeatability of tuning the position controller's proportional element is the worst of the three target parameters, while the velocity controller's proportional element has the best tuning repeatability. The testers generally increased all the available controller tuning parameters when the mass on the axis was increased. After the test event, the standardized controller performance measurement test described in Section 4.1 was run on each of the tuning parameter sets and their respective mass configurations. The ISE results from the test are also gathered in Appendix C.

Table 6.1 *Mean and standard deviation of human tuning performance*

	Velocity controller	Position controller
ISE mean	194.2	$1.094 \cdot 10^{-2}$
ISE σ	83.7	$1.416 \cdot 10^{-2}$

As outlined in the test description (Section 4.3), the different-mass ISE results for the velocity and position controllers can be combined to have single mean and standard deviation results for each of the controllers. Further justification for combining the results is presented in Appendix D. The calculated mean and standard deviation values for the performance results are given in Table 6.1. The values from the velocity controller and position controller cannot be compared due to their different units. At this point, the human performance results cannot be analysed in-depth, as there are no reference points in which to compare them. However, especially the position controller's standard deviation is concerning, as it is larger than the mean performance result. When it comes to errors in the performance measurement, there is always some measurement noise, especially in the velocity data, as it has been derived from discrete axis position data. Other sources of error include small variations in the friction of the test axis, caused by wear and temperature variation of the axis' linear bearings. Additionally, temperature and random measurement noise affect the feedback device of the system. To mitigate these sources of errors as much as possible, the performance tests were conducted on the same day as the test subjects found the tuning parameters.

6.2 Velocity model-finding experiment behavior

The test results of the model-finding experiments for the auto-tuning methods are presented next. Their implementations are described in Chapter 5. If a model-finding experiment is unusable with the velocity or position controller of the target system, it makes no sense further measure and calculate the performance of the tuning parameters produced by an auto-tuning method that utilizes the experiment. In the OptoDrive, the velocity controller is closer to the controlled process than the position controller. This can be seen from the OptoDrive's controller structure, where the velocity controller gets its setpoint from the position controller in normal operation (see Section 3.2). Therefore, the model-finding experiments were tested first on the velocity controller.

The behavioral analysis for the model-finding experiments for tuning the velocity controller is divided into two sections, first analysing the axis velocity and then analysing the axis position during the experiments. The axis velocity during the experiments is first analysed individually for each of the experiments. Afterwards, the axis position during the experiments is analysed for all three experiments together. The purpose of analysing the servo axis velocity is to see whether the experiments have been properly implemented and if the desired axis behavior can be achieved with the target axis. The purpose of the position analysis is to have a more easily understandable, objective plot of the servo axis behavior during the experiments. The axis position information is therefore a matter of compatibility with OptoFidelity's robots and the auto-tuning use-case.

Open-loop setpoint response experiment

First, an implementation of the open-loop setpoint response experiment was created, according to its functional definition in Section 5.3. The velocity controller output, axis torque, had to be to roughly 40 percent of the maximum available torque to reach a stable velocity on the axis. The lowest stable velocity was roughly 100 $^{\mathrm{mm}}/_{\mathrm{s}}$, seen in the resulting process response plot, in Figure 6.1. It resembles the ideal response plot given in Figure 2.3, which shows that the servo drive is properly controlled and configured in the experiment. The velocity in the figure begins to taper off towards the end of the trajectory, likely due to varying friction on the servo axis. Lower torques on an axis had the tendency to allow the axis to stop entirely. The light mass plate was installed on the axis during this testing. The region of zero axis velocity in the beginning of the measured data is a result of beginning the data sampling before applying the setpoint.

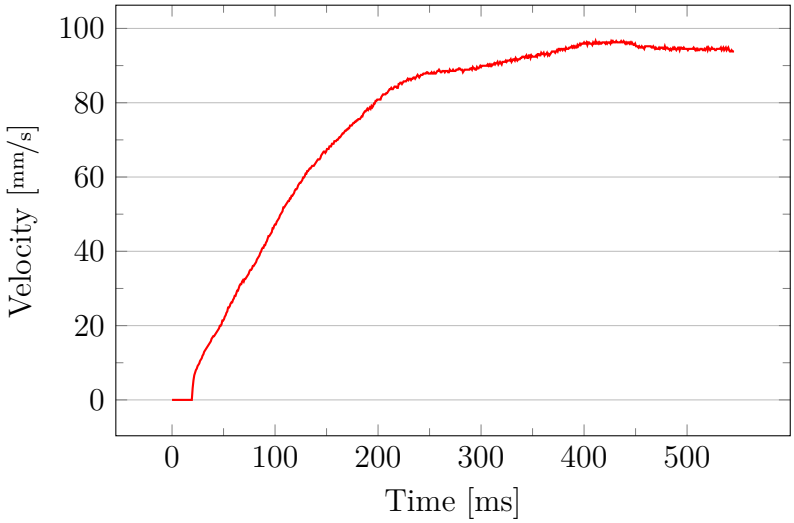

Figure 6.1 *Axis velocity during the velocity controller's open-loop experiment*

The model parameters extracted from the first test (Figure 6.1) are given in Table 6.2. In the test, the torque of the servo motor was 40 percent of the maximum torque, and its stable velocity was 14.6 percent of the maximum velocity. This equates to a process gain K_{pg} of 0.365 (Formula 2.12). A tangent drawn on the steepest slope of the curve intersects the time axis at time 17.4 ms. 63 percent of the process value was reached at time 129 ms, from the beginning of the test. This results in a process time constant τ_c of 111.6 ms. Based on the near-immediate initial rise of the process value, the process does not have much dead time. The calculated dead-time was constantly four controller cycles over all the conducted tests. With the 2500 Hz controller frequency, this equates to θ_d of 1.6 ms. Considering the place of the dead-time in the denominator of the Cohen-Coon method's formulae in Table 2.2 and the aforementioned sensitivity of the method to the friction on the servo axis, there is potential for error in the calculation of the resulting controller tuning parameters. Applying the acquired parameters to the Cohen-Coon method's formulae (Table 2.2 and Formula 2.10) results in tuning parameters $K_\mathrm{p} = 172$ and $K_\mathrm{i} = 193$. The parameters were applied to the formulae as seconds, but are shown here as milliseconds for readability. When comparing these to the tuning parameters made by humans for the light mass plate configuration (Appendix C), the parameters seem viable for use.

Table 6.2 *Model parameters extracted from the velocity controller's open loop experiment*

	K_{pg}	τ_c [ms]	θ_d [ms]
Result	0.365	111.6	1.6

Closed-loop setpoint response experiment

The closed-loop setpoint response experiment also produced a near-textbook response with no major issues visible in the plot, given in Figure 6.2, with the light mass plate on the axis. Data gathered during the experiment was limited in its length with the purpose of better illustrating the experiment's overshoot and undershoot in the response plot. The original data is 250 ms long and continues a stable response, with the same measurement noise that can be seen in the plot. The experiment was implemented according to the definition in Section 5.4. Its nominal behavior is given in Figure 2.4. The measurement noise that is seen in the plot raises some concern. Although the amplitude of the noise is low compared to the amplitude of the process peaks, some measurement error in the resulting peak times and relative amplitudes can be expected. These measurement errors, likely resulting from the derivation of the feedback device's data, can be expected to cause some degradation of the repeatability of the experiment and thus the resulting controller tuning parameters.

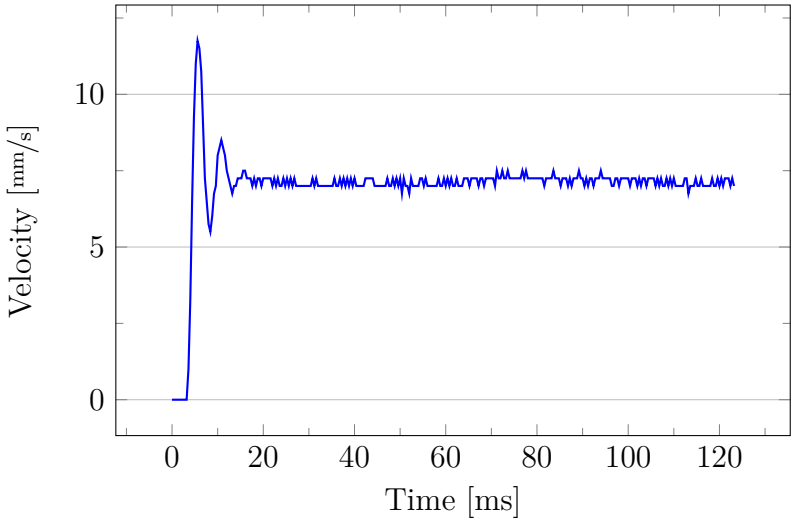

Figure 6.2 *Axis velocity during the velocity controller's closed loop experiment*

This test process response (Figure 6.2) was analysed to find the parameters for the auto-tuning methods' use. The process gain K_{p0} during the experiment was 515. The time from setpoint application to the initial peak t_p was analysed first, with a result of 3.1 ms. The controller setpoint Δy_s was 10 mm/s during the experiment, and as the process settled at the Δy_∞ of 7.38 mm/s, the relative process steady state change b is 0.738 (Formula 2.13). Finally, as the peak process change Δy_p was 12.0 mm/s, the relative overshoot A_0 is 0.626 (Formula 2.14). These model parameters are summarized in Table 6.3.

Table 6.3 *Model parameters extracted from the velocity controller's closed loop experiment*

	K_{p0}	t_p [ms]	b	A_o
Result	515	3.1	0.738	0.626

These model parameters can be applied to the tuning parameter calculation formulae, presented in Section 2.3.4 for the closed-loop SIMC method and in Section 2.3.3 for the setpoint overshoot method. From these model parameters, the setpoint overshoot method produces tuning parameters $K_{\mathrm{p}} = 229$ and $K_{\mathrm{i}} = 299$ (Formulae 2.15, 2.16, 2.17, and 2.10). The closed-loop SIMC method produces tuning parameters $K_{\mathrm{p}} = 293$ and $K_{\mathrm{i}} = 512$ (Formulae 2.15, 2.18, 2.19, 2.20, 2.21, 2.22, 2.23, and 2.10). These tuning parameters were calculated with the recommended adjustment parameter τ_t values of 1 for the setpoint overshoot method, and θ_d (calculated during analysis) for the closed-loop SIMC method. Considering the parameters made by humans for the light mass plate (Appendix C), both parameter sets are likely to work with the system.

Ultimate gain experiment

Finally, the ultimate gain experiment was implemented on the system, based on the description in Section 5.2. The test system's response during the ultimate gain experiment, with the light mass plate installed on the system, is given in Figure 6.3. Oscillation on the axis is initiated with a single 3.1 mm step in the position setpoint of the servo drive, which causes a 30 $^{\mathrm{mm}}/_{\mathrm{s}}$ spike in the velocity setpoint. The spike in the setpoint quickly settles to zero and the velocity controller transitions to an oscillatory state, provided that its proportional gain is large enough. Running the response through an FFT operation results in a spike at the frequency of the system oscillation. The result of the FFT operation is given in Figure 6.4. The oscillation spike can be reliably distinguished from measurement noise and the setpoint change that induced the initial oscillation. Due to the sample rate of the process data, the frequency data produced by FFT may have some degree of error in it. In the initial tests, the axis oscillation frequency was detected with an under 1 Hz repeatability.

After ramping up the K_{p} of the velocity controller, so that oscillation is found on the system, the resulting FFT plot (Figure 6.4) was analysed to find the frequency of the oscillation. In the test, the oscillation was first detected with a controller gain K_{p} of 1394. Based on the FFT data, the oscillation frequency of the test data is 207.9 Hz. This equates to an oscillation period P_{osc} of 0.004 81 s. These model parameters are summarized in Table 6.4.

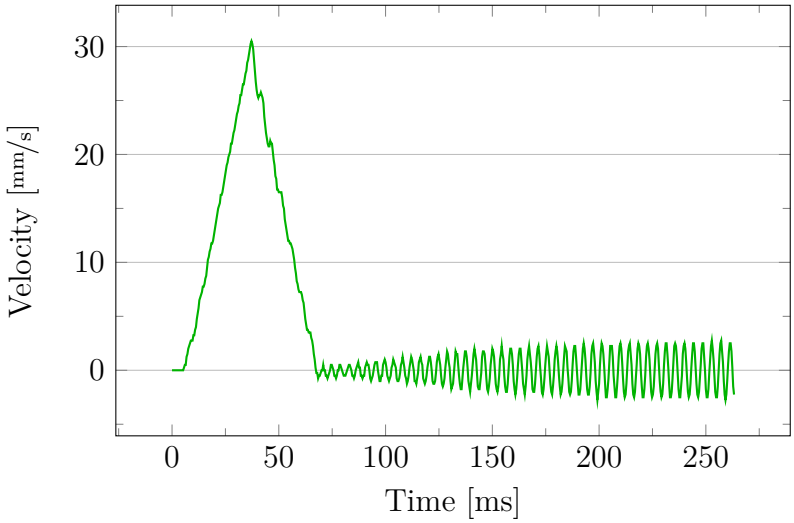

Figure 6.3 *Axis velocity during the velocity controller's ultimate gain experiment*

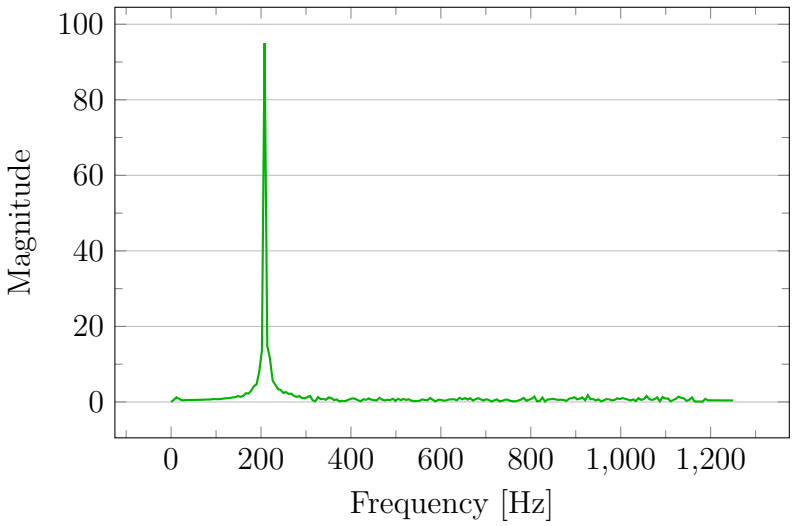

Figure 6.4 *FFT result calculated from the axis velocity data during the ultimate gain experiment*

Table 6.4 *Model parameters extracted from the velocity controller's ultimate gain experiment*

	K_{p_u}	P_{osc} [s]
Result	1394	0.00481

The resulting test tuning parameters for the light mass plate are calculated from the formulae presented in Table 2.1. Depending on the Ziegler-Nichols method's variant, different tuning parameters are produced from the model parameters. The "classic" variant produces velocity controller tuning parameters $K_\mathrm{p} = 627$ and $K_\mathrm{i} = 249$. The "some overshoot" variant produces tuning parameters $K_\mathrm{p} = 460$ and $K_\mathrm{i} = 416$. Lastly, the "no overshoot" variant produces tuning parameters $K_\mathrm{p} = 279$ and $K_\mathrm{i} = 416$. Upon comparing these to the light mass plate velocity controller tuning parameters made by humans (Appendix C), the "classic" variant produced significantly higher proportional gain than humans. The other variants produced comparable parameters to humans, so these tuning parameters are likely to work with the test axis.

Axis position during the experiments

After implementing the experiments and inspecting their behavior in regard to the velocity of the servo axis, the position of the axis during the experiments was plotted to assess the methods' compatibility with OptoFidelity's auto-tuning use case. The experiments' position behavior is shown in Figure 6.5. From the position plot, it is apparent that the open-loop setpoint response experiment is not suitable for tuning servo axes in measurement robotics for two reasons. First, it requires a significant amount of distance on the servo axis during the experiment, and the space is not always available. Axes as small as 5 mm in length can be subjects for auto-tuning. Secondly, the large and uncontrolled movement of the axis during auto-tuning is not acceptable for a measurement robot, where fragile and bulky instruments can be mounted to it. The closed-loop and the ultimate gain experiments appear to be behave suitably to be considered for use. They do not travel far and their behavior is deterministic.

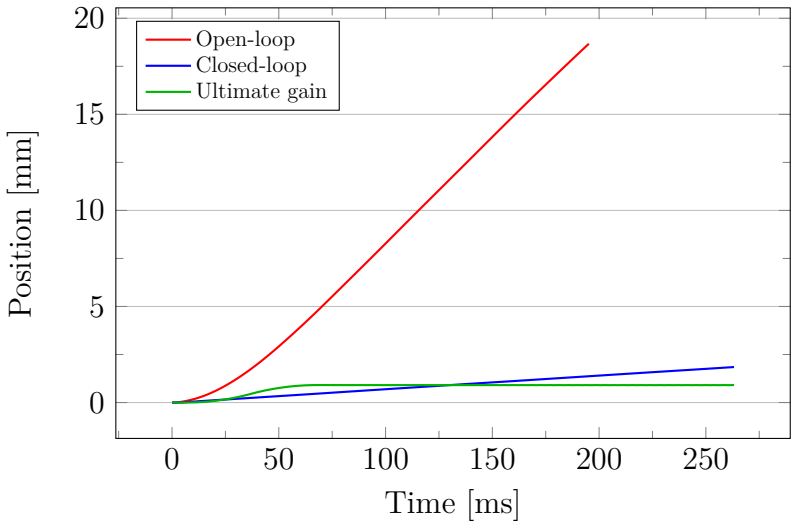

Figure 6.5 *Axis position during the velocity model-finding experiments*

Summarising the behavior of the velocity model-finding experiments, the open-loop setpoint response experiment was found to be unsuitable for the targeted use case, but the closed-loop setpoint response experiment and the ultimate gain experiment show promise. They allow the use of the setpoint overshoot, the closed-loop SIMC, and the Ziegler-Nichols controller auto-tuning methods for tuning the OptoDrive's velocity controller. During initial testing, all of the tested auto-tuning methods produced tuning parameters with a similar magnitude to the parameters produced by humans. The time it took for the experiments to finish was under one minute for all the experiments.

6.3 Velocity controller auto-tuning performance

Continuing from the experiments in the previous section, auto-tuning performance of the setpoint overshoot (Section 2.3.3), the closed-loop SIMC (Section 2.3.4), and the Ziegler-Nichols (Section 2.3.1) controller auto-tuning methods was measured with the test system. Upon initial testing of the produced test controller tuning parameters, it was found that the Ziegler-Nichols method produced oscillatory controller behavior with its "classic" variant, and stable controller behavior with its "some overshoot" and "no overshoot" variants. From the stable variants, the "some overshoot" produced better performance values, so it was selected for further testing. The recommended adjustment parameter τ_t values of 1 for the setpoint overshoot method, and θ_d for the closed-loop SIMC method were used during testing.

As the position controller of the system is required to be active for the performance measurement test (Section 4.1), its tuning parameter K_p was set to 1800 for the entire duration of velocity controller testing. According to the tuning parameters made by humans (Appendix C), this value should work with all the mass configurations, and yield satisfactory position controller behavior without pushing it to its limits. It is likely that different position controller gains would cause the velocity controller to perform differently due to controller coupling. However, because of time and research scope constraints, this single chosen value will be used during these tests.

The auto-tuning methods were tested according to the testing plan described in Section 4.2. The test does 30 controller tuning cycles for an auto-tuning method, with ten rounds for each of the three available axis mass configurations. The ISE performance of the resulting tuning parameters is then measured and calculated using the performance measurement test (Section 4.1). A box plot was drawn from each of the 30 performance indices to visualize the methods' performance and its distribution. Box plots for each of the three tested tuning methods are given in Figure 6.6. The produced tuning parameters and recorded performance values are given in Appendix (E). According to the box plots, all the performance results are

biased towards better performance. Interestingly, despite their identical method of model-finding, the setpoint overshoot method and the closed-loop SIMC method have significantly different repeatability of performance. The closed-loop SIMC method is more repeatable according to the test, and produces the best performing controller tuning values from the three tested methods, overall. All of its tuning parameters' performance results are better than three quarters of the second-best method's, the Ziegler-Nichols "some-overshoot" variant's tuning parameters.

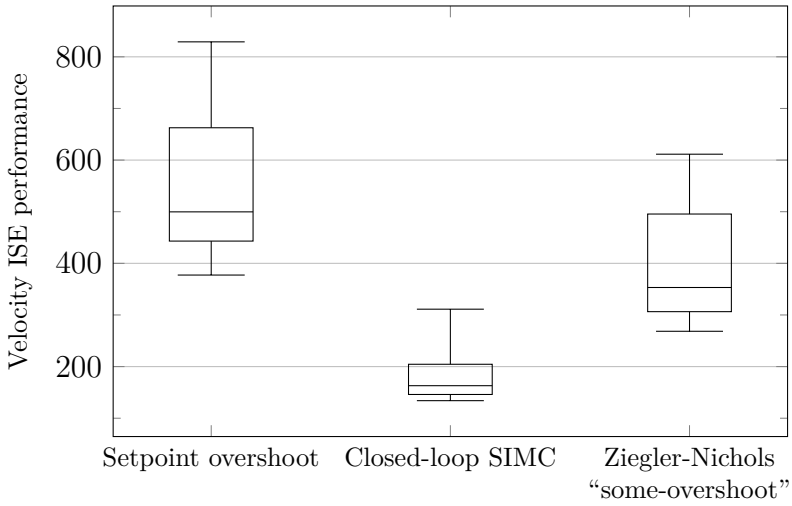

Figure 6.6 *Velocity controller auto-tuning methods' ISE performance distribution*

Mean values and standard deviation values of the performance results (Appendix E) were then calculated to allow the auto-tuning methods to be compared to humans. The calculated results, along with the human performance baseline, are given in Table 6.5. The results reflect the findings made when analysing the box plot visualization of the performance results. The average ISE result for the best tested velocity controller auto-tuning method, the closed-loop SIMC method, is 91 percent of the humans' equivalent result. Thus, there is less tracking error in the controller operation when it is tuned by the closed-loop SIMC method, than when it is tuned by humans. The standard deviation of the closed loop SIMC method's performance is 58 percent of the humans' standard deviation. As the standard deviation of performance describes tuning repeatability, based on the results, the closed-loop SIMC method is more repeatable than humans in tuning the velocity controller. The second-best auto-tuning method, the Ziegler-Nichols "some-overshoot" variant, has 96 percent larger ISE mean result than humans, so humans achieved significantly better tuning results. It also has 30 percent larger standard deviation of performance than humans. Based on these results, the closed-loop SIMC method should be used for auto-tuning the velocity controller of the OptoDrive.

Table 6.5 *Velocity controller auto-tuning methods' ISE performance mean and standard deviation*

	Setpoint overshoot	Closed-loop SIMC	Ziegler-Nichols "some-overshoot"	Human result
ISE mean	529.1	176.7	379.8	194.2
ISE σ	121.9	48.7	109.0	83.7

6.4 Position model-finding experiment behavior

After testing the velocity controller, the model-finding experiments were tested for the position controller. The position controller has two major technical differences compared to that of the velocity controller. First, it is a P-controller instead of a PI-controller and secondly, it does not directly control an actuator, but rather gives the velocity controller setpoints. The behavior of the model-finding experiments is more straightforward to analyse for the position controller. This is because the experiments' behavior and the system's behavior can be analysed from the same axis position plot.

As discussed in Section 5.3, the open-loop setpoint response experiment is not usable with the position controller. No such limitations were faced during the literature review or the implementation work for the closed-loop setpoint response experiment and the ultimate gain experiment. However, while the closed-loop setpoint response experiment was successfully written and tested for the position controller, the behavior of the system was not ideal during the experiment. A plot of the axis position during the closed-loop setpoint response experiment is given in Figure 6.7. The light mass plate was installed on the axis during testing. The setpoint change step used in the experiment was 1 mm. An overshoot of 10 percent from the stable process state, required by the experiment, could not be achieved with a well tuned velocity controller without extreme axis oscillation. The best velocity controller tuning parameters made by humans (Appendix C) and the auto-tuning methods (Closed-loop SIMC, Appendix E) were tested. When the velocity controller tuning parameters were purposefully set incorrectly, an overshoot in the axis position during the experiment was achievable without axis oscillation, but this undermines the purpose of the model-finding experiment. After all, it should find the model of the system that will actually be used, and not of a system that has been temporarily modified for the experiment.

The extreme axis oscillation, that was observed during the closed-loop setpoint response experiment (Figure 6.7), caused the servo drive to output high electrical current to the system's motor during testing. This, in turn, caused the servo drive

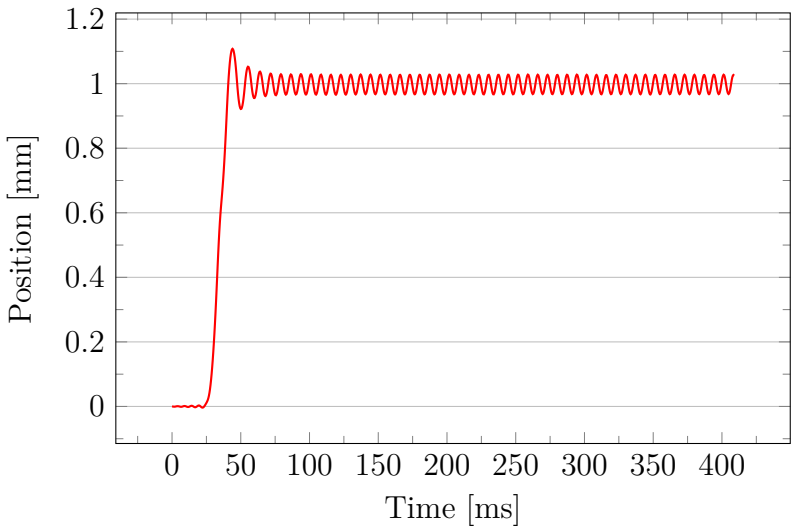

Figure 6.7 *Axis position during the position controller's closed loop experiment*

to go into thermal protection mode. From this system response, it was determined that the closed-loop setpoint response experiment is unsafe for use with the position controller of the OptoDrive.

Plots gathered during the testing of the closed-loop setpoint response experiment show promise regarding the ultimate gain experiment for the position controller, despite their failure for their intended purpose. Sustained oscillation was found to be achievable on the servo axis. The K_p of the controller was reduced from the setpoint response experiment to find the actual critical gain where the oscillation begins. Axis behavior during the ultimate gain experiment, with critical proportional controller gain, is shown in Figure 6.8. In the beginning of the plot, the 1 mm step used to initiate possible axis oscillation can be seen. After the step, the axis has a tidy sustained oscillation. This produces an easily detectable oscillation peak in a resulting FFT plot, given in Figure 6.9. The ultimate gain experiment did not trigger the thermal protection feature of the servo drive.

The model parameters for the Ziegler-Nichols auto-tuning method were then gathered from the results of the ultimate gain experiment. The position controller of the OptoDrive consists of a single proportional control element. Only the ultimate gain K_{p_u} of the experiment is required to calculate a gain for it (Table 2.1). Its value in the experiment was 4706, when the light mass plate was installed on the system, and the best auto-tuned velocity controller tuning parameters were used (closed-loop SIMC, Appendix E). For reference, the system's oscillation frequency during the experiment was 130.6 Hz, which equates to an oscillation period P_{osc} of 0.007 66 s. The resulting K_p, using the Ziegler-Nichols "classic" variant, is 2353 for the light mass plate configuration. Compared to the tuning parameters made by humans (Appendix C), the resulting gain is slightly high, but possibly viable for use.

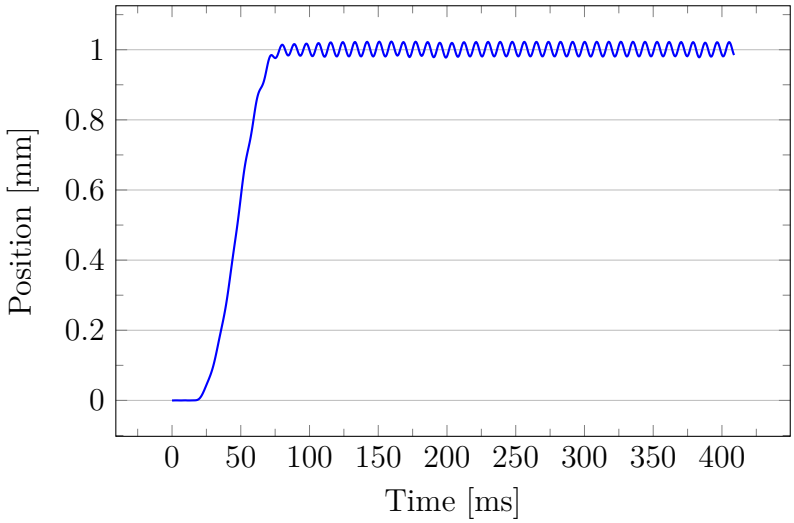

Figure 6.8 *Axis position during the position controller's ultimate gain experiment*

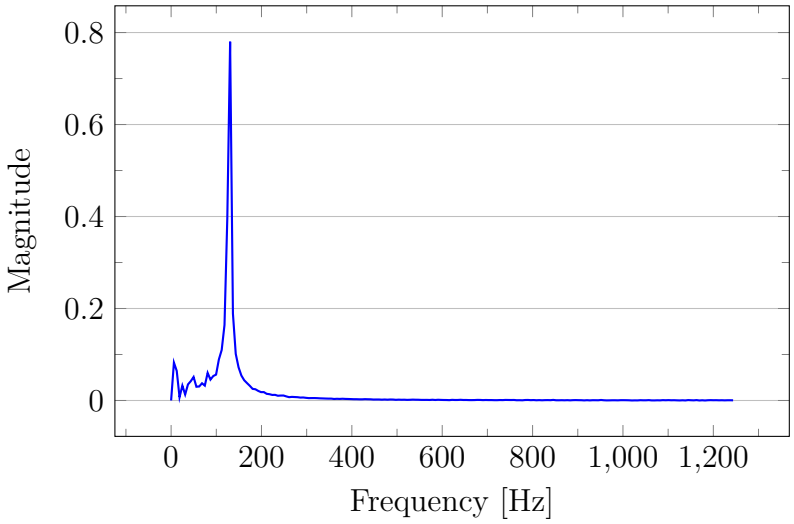

Figure 6.9 *FFT result calculated from the axis position data during the ultimate gain experiment*

Table 6.6 *Model parameters extracted from the position controller's ultimate gain experiment*

	K_{p_u}	P_{osc} [s]
Result	4706	0.00766

Out of the three tested system model-finding methods for the position controller, only the ultimate gain method was found to be suitable for use. The open-loop setpoint response experiment is inherently not suitable for use, and its closed-loop variant is dangerous for the system hardware. The reason why the closed-loop setpoint response experiment requires such a high proportional gain to produce the required overshoot may be that the dead time of the controlled process is very low. Therefore, the system's controller does not have time to uncontrollably ramp up its output without the process catching up. If in the future the OptoDrive is used with a differently behaving system, such as a belt-driven robot with feedback sensor on the axis and not directly coupled to the motor, the closed-loop setpoint response experiment could be tested on it. Also, making the axis oscillate is not the ideal model-finding experiment behavior, but can be acceptable if the oscillation is only momentary. The closed-loop setpoint response experiment makes the process oscillate long before sufficient overshoot is achieved, while the ultimate gain experiment requires only one test with sustained oscillation. Similarly to the model-finding experiments for the velocity controller, it took less than one minute for the experiments to finish. Consequently, the Ziegler-Nichols controller auto-tuning method, that utilizes the ultimate gain experiment, is the only auto-tuning method whose position controller tuning performance is tested.

6.5 Position controller auto-tuning performance

Testing the auto-tuning methods for the position controller follows a similar testing and analysis methodology that has already been conducted for the velocity controller. However, the cascade arrangement of the OptoDrive poses an issue for measuring the position controller's performance. If the velocity controller is unable to track its setpoint accurately, it induces errors in the axis position regardless of the position controller's behavior. This results in reduced observed position controller performance. Due to this unfortunate coupling of the controllers, the position controller auto-tuning performance was tested with velocity controller tuning parameters, made with the closed-loop SIMC method and the Ziegler-Nichols "some-overshoot" variant. The closed-loop SIMC method is included, because it produced the best auto-tuning results for the velocity controller. This will minimize the position control error, that is caused by the velocity controller. The Ziegler-Nichols "some-overshoot" variant is included in testing because it was considered interesting to see how the Ziegler-Nichols method performs alone. After all, it is the only auto-tuning method that is compatible with both controllers.

The velocity controller tuning parameters were selected from the auto-tuning test results (Appendix E). The best performing K_p and K_i tuning parameters were

chosen for both selected velocity controller tuning methods, and for each of the mass configurations. The selected parameters, that were used during position controller auto-tuning and testing, are given in Table 6.7.

Table 6.7 *Velocity controller parameters used during position controller performance testing*

	Closed-loop SIMC		Ziegler-Nichols "some-overshoot"	
	K_p	K_i	K_p	K_i
No mass plate	192	338	342	308
Light mass plate	294	478	470	416
Heavy mass plate	328	574	552	484

The Ziegler-Nichols "classic" position controller auto-tuning method was tested according to the description in Section 4.2. The obtained tuning parameters and ISE results are given in Appendix F. The ISE results for the mass configurations were again combined, like with human and velocity controller auto-tuning method testing (see Appendix D for justification). A box plot visualization of the Ziegler-Nichols method's position controller tuning performance was drawn to analyse the distribution of the performance results. The visualization is given in Figure 6.10. As before with the velocity controller, the results are skewed towards better performance (lower ISE). Based on the results, the selection of the velocity controller's tuning parameters affects the position controller's performance heavily. All of the results with the closed-loop SIMC velocity controller tuning parameters are better than the median result that is obtained with the Ziegler-Nichols "some-overshoot" velocity controller tuning parameters. Better velocity controller tuning parameters also seem to improve the spread of the position controller's auto-tuning performance results.

The mean and standard deviation of the obtained performance indices (Appendix F) are given in Table 6.8. The corresponding performance results for humans (Appendix C) are also included in the table. According to the performance results, the Ziegler-Nichols has a large performance advantage over humans when tuning the position controller of the OptoDrive, regardless of the velocity controller tuning parameters that were used. When the velocity controller was tuned with the closed-loop SIMC method, the Ziegler-Nichols "classic" variant produced 82 percent better ISE mean performance than humans. The humans' standard deviation of performance was almost 58 times larger in this case. This is logical considering the observations done during the manual tuning experiment, where the position controller tuning parameters made by humans varied by a significant amount. When tuning the velocity controller with the Ziegler-Nichols "some-overshoot" method, the Ziegler-Nichols "classic" method produced 36 percent worse mean performance, and 77

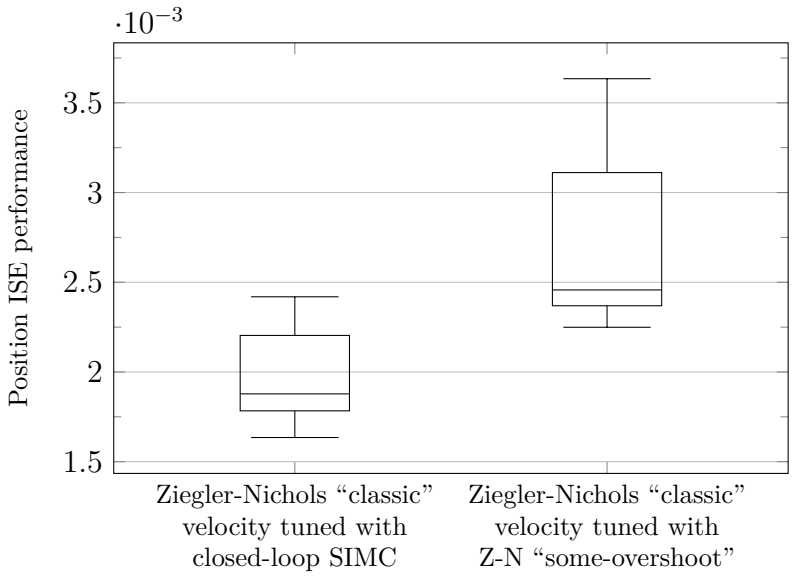

Figure 6.10 *Position controller auto-tuning methods' ISE performance distribution*

percent worse standard deviation of performance than in the optimal case (closed-loop SIMC velocity tuning). Based on these results, to obtain best performance on the OptoDrive, the Ziegler-Nichols "classic" should be used to tune the position controller, and the closed-loop SIMC should be used to tune the velocity controller.

Table 6.8 *Position controller auto-tuning method's ISE performance mean and standard deviation*

	Ziegler-Nichols "classic" velocity tuned with closed-loop SIMC	Ziegler-Nichols "classic" velocity tuned with Z-N "some-overshoot"	Human result
ISE mean	$1.953 \cdot 10^{-3}$	$2.661 \cdot 10^{-3}$	$1.094 \cdot 10^{-2}$
ISE σ	$2.462 \cdot 10^{-4}$	$4.346 \cdot 10^{-4}$	$1.416 \cdot 10^{-2}$

6.6 Optimized Ziegler-Nichols

The tested controller tuning methods' formulae are flexible in their tuning parameter calculation. This opens up an opportunity to optimize them for the target process. The setpoint overshoot and the closed-loop SIMC methods allow for the adjustment of the resulting tuning parameters with their τ_t parameters (see Sections 2.3.3 and 2.3.4). For the Ziegler-Nichols method, there are many different formulae available for calculating tuning parameters from the system's ultimate gain and oscillation period. A few of them are presented in Table 2.1. However, using trial and error to seek for the optimal adjustment parameters or other multipliers is not desirable, especially as the tuning methods could be used in customer premises, where the speed

and the repeatability of tuning is critical. The Ziegler-Nichols controller auto-tuning method was found to be suitable for tuning both controllers of the OptoDrive, so it was selected for optimization testing. Optimization effort was not put into the best controller auto-tuning method for the velocity controller, the closed-loop SIMC method, because of time constraints. Optimizing all the tested controller auto-tuning methods is something to research as future work.

Optimization method

For the optimization, a method for fine-tuning the multipliers of the Ziegler-Nichols method was devised in cooperation with Eero Heinänen, the developer of the second part of the OptoDrive's tuning sequence (see Section 1.1). Essentially, the method, referenced as the Koljonen-Heinänen -method or the KH-method, produces new variants of the Ziegler-Nichols method's tuning parameter calculation formulae. It utilizes a model–free Adaptive Genetic Algorithm (AGA) in its optimization process. Genetic algorithms optimize the value of a fitness function by manipulating a set of variables over several iterations, and they have previously been used for tuning PID controllers (see Porter and Jones 1992). The fitness function is used to calculate the goodness of the variable set that is being optimized (Robinson and Rahmat-Samii 2004). While these kinds of iterative methods can be used for controller tuning, it takes a long time for them to reach the optimum result (for example, 20 iterations with population size 16, a total of 320 tuning parameter sets to test in Porter and Jones 1992). Therefore, optimizing the Ziegler-Nichols, or another auto-tuning method to produce good results quickly is desirable. The advantage of the AGA over a regular genetic algorithm is that it has a better probability of finding the global optimum, rather than a local optimum of the fitness function (Srinivas and Patnaik 1994, Mahmoodabadi and Nemati 2016). It was selected for use, because at the time of testing, it was the best performing optimization algorithm implemented by Eero Heinänen in his thesis experiments.

The position controller's ISE performance result is used as the fitness function in the optimization process, to optimize the servo drive's ability to follow set trajectories. It is measured and calculated with the controller performance test, described in Section 4.1. The method first finds the most optimal velocity and position controller tuning parameters for a given process using the AGA. Then the model finding experiment of the auto-tuning method that is being optimized is run ten times for both controllers. This repetition is done in an effort to minimize any measurement uncertainty. As this test focuses on the Ziegler-Nichols method, the ultimate gain experiment is used (see Section 2.3.1 and Algorithm 2.1). The mean values of the resulting model parameters, here the ultimate gains K_{p_u} and controller oscillation periods P_{osc}, are then calculated. Finally, the optimal tuning parameter calculation multipliers for

the Ziegler-Nichols method are solved from the optimal tuning parameters and mean system model, and applied to similar equations as presented in Table 2.1.

Application of the optimization method

In practice, the devised KH-method was used to find new Ziegler-Nichols tuning parameter calculation formulae for the velocity and the position controllers of the target axis in this thesis, with the light mass plate configuration. The tuning parameters for the velocity and the position controller of the OptoDrive were optimized simultaneously. The AGA found that the most position ISE-optimal controller tuning parameters are $K_\mathrm{p} = 2257$ for the position controller, and $K_\mathrm{p} = 424$ and $K_\mathrm{i} = 341$ for the velocity controller. The ISE results were 143.3 for the velocity controller and 0.000899 for the position controller. The mean model parameters were found to be $K_{\mathrm{p}u} = 3368$ for the position controller, and $K_{\mathrm{p}u} = 1368$ and $P_{osc} = 0.0048$ for the velocity controller, calculated from the specified ten test runs. The integral controller element is not used for the position controller, and therefore it does not require the integral gain parameter. The oscillation period of the position controller was not measured, as it is only required for calculating the integral time (see Table 2.1).

Table 6.9 *Formulae produced with the KH-method for the Ziegler-Nichols method*

Variant	Controller	Controller type	K_p	T_i
Ziegler-Nichols "K-H"	Position	P	$0.67K_{\mathrm{p}u}$	–
	Velocity	PI	$0.31K_{\mathrm{p}u}$	$P_{osc}/1.65$

These given optimal tuning parameters and mean model parameters were used to solve the new formulae for the Ziegler-Nichols method. The resulting formulae are given in Table 6.9, referenced as the Ziegler-Nichols "K-H" variant. When compared to the formulae found in literature (see Table 2.1), the new variant produces higher proportional gains for the position controller. However, it produces lower proportional and integral gains than the best tested variant (the "some-overshoot", see Section 6.3) for the velocity controller. The higher resulting integral time equates to lower resulting integral gain (Formula 2.10). As described, the light mass plate was installed on the axis for the entire duration of the optimization. The same Ziegler-Nichols "K-H" variant will be used to auto-tune the OptoDrive with all the mass configurations of the test axis to see how universal the new multipliers are.

Performance results for the velocity controller

To evaluate the optimization results, the Ziegler-Nichols "K-H" variant was first tested on the velocity controller of the OptoDrive. The standardized auto-tuning

method performance test (see Section 4.2) was used to gauge its performance, with the position controller's K_p set to 1800, which was previously used when testing the velocity controller auto-tuning methods (Section 6.3). The resulting box plot is given in Figure 6.11. The results for the closed-loop SIMC method and the "some-overshoot" variant of Ziegler-Nichols method are given in the plot for reference. The obtained controller tuning parameters and ISE results are given in Appendix E.

Upon comparing the distribution of the KH-optimized Ziegler-Nichols variant's performance results for tuning the velocity controller to other auto-tuning methods, the performance of the Ziegler-Nichols "K-H" variant is outstanding. The optimized Ziegler-Nichols "K-H" variant surpassed the previously best closed-loop SIMC method in performance when tuning the velocity controller. The Ziegler-Nichols "K-H" variant has significantly less variance in the performance results than the "some-overshoot" variant, probably because of more repeatable controller behavior. The worst velocity controller tuning performance obtained with the Ziegler-Nichols "K-H" variant is better than the best result of the "some-overshoot" variant. This gives credibility to the KH-method, as the Ziegler-Nichols "K-H" variant seems to perform well with all three mass configurations on the test axis, despite being obtained with a single mass configuration.

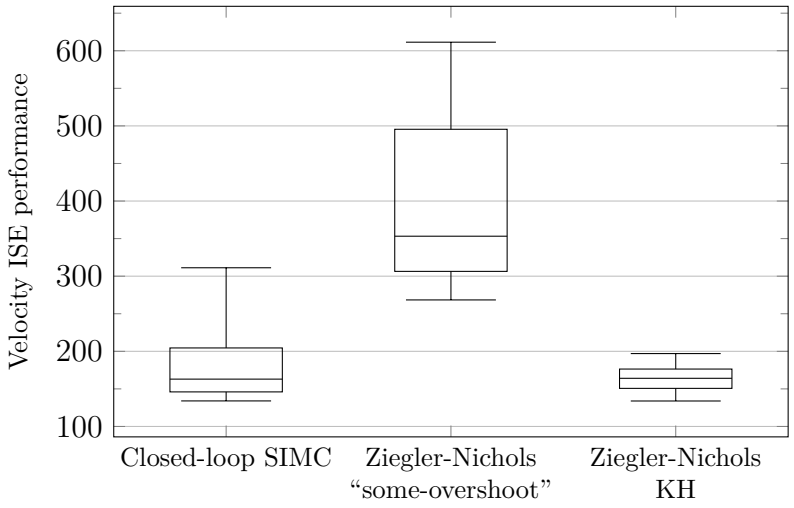

Figure 6.11 *KH–optimized Ziegler-Nichols auto-tuning method's performance distribution with the velocity controller*

The mean and standard deviation values for the Ziegler-Nichols "K-H" variant's performance results are presented in Table 6.10. The mean performance result of the "K-H" variant is about 93 percent of the closed-loop SIMC's performance result. It's standard deviation is about 35 percent of the closed-loop SIMC's results. The mean Ziegler-Nichols "K-H" variant's ISE performance result for the velocity controller is about 84 percent of the ISE performance that humans achieved during testing, with only 20 percent of humans' standard deviation of ISE performance. It is the best auto-tuning method that has been found for the velocity controller of the OptoDrive.

Table 6.10 *Ziegler-Nichols KH velocity controller auto-tuning method's ISE performance*

	Ziegler-Nichols "K-H"	Closed-loop SIMC	Ziegler-Nichols "some-overshoot"	Human result
ISE mean	163.7	176.7	379.8	194.2
ISE σ	17.1	48.7	109.0	83.7

Performance results for the position controller

After the velocity controller, the Ziegler-Nichols "K-H" variant was tested on the position controller of the OptoDrive. Similarly to the previous position controller testing (Section 6.5), the selected velocity controller tuning parameters are significant for the position controller's performance. When selecting the velocity controller tuning parameters, the best parameters for each of the three mass configurations, based on ISE performance were used from the results in Appendix E. First, the position controller was tuned with the Ziegler-Nichols "K-H" variant, paired with Ziegler-Nichols "K-H" variant -tuned velocity controller. Out of interest, the position controller was tuned again with the Ziegler-Nichols "K-H" variant, but paired with closed-loop SIMC -tuned velocity controller. Last, the Ziegler-Nichols "classic" position controller tuning method was re-tested, paired with Ziegler-Nichols "K-H" -tuned velocity controller. The tuning parameters and performance results from these tests are presented in Appendix F.

The resulting box plots are given in Figure 6.12. The previously measured Ziegler-Nichols "classic" & closed-loop SIMC -pair's results are given in the figure as reference. From the box plots, it is apparent that the Ziegler-Nichols "K-H" variant performs well for tuning the position controller. Re-tuning the position controller with the the Ziegler-Nichols "classic", with the velocity controller tuned with the Ziegler-Nichols "K-H", produced more spread in the performance than was previously observed with the closed-loop SIMC method. However, its median result is still better with the Ziegler-Nichols "K-H" than the closed-loop SIMC -produced velocity controller tuning parameters.

The mean and standard deviation values of the performance results are presented in Table 6.11. When both controllers were tuned with the Ziegler-Nichols "K-H" variant, the mean ISE performance is about 10 percent of what humans achieved. The standard deviation for the Ziegler-Nichols "K-H" -pair is even better, 1.4 percent of the humans' result. The other velocity and position controller tuning method pairs produced worse results than the Ziegler-Nichols "K-H" -pair, but still better results than humans, by a good margin.

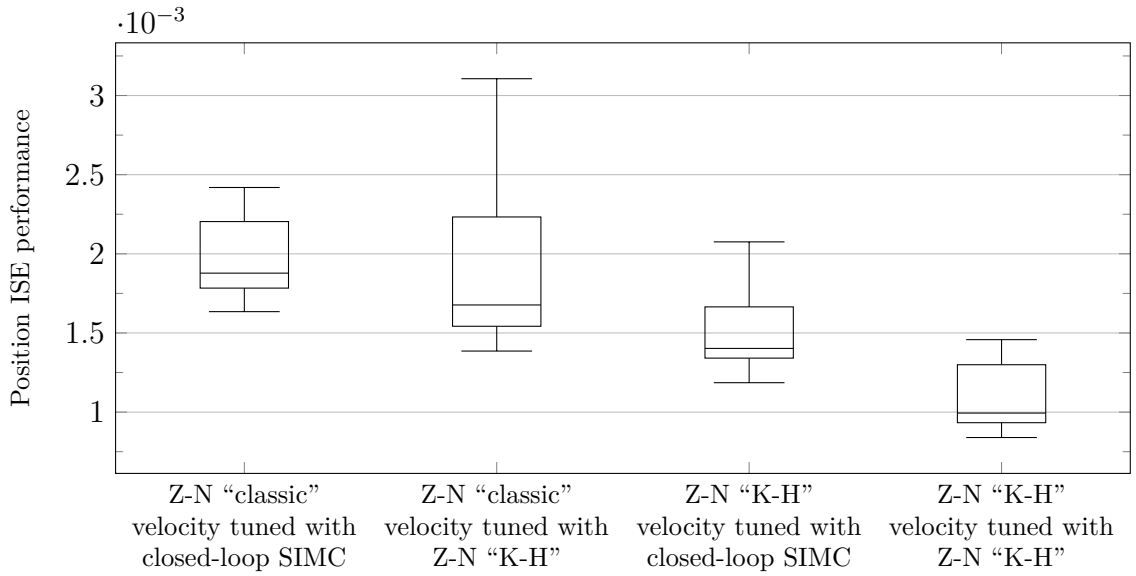

Figure 6.12 *KH–optimized Ziegler-Nichols auto-tuning method's performance distribution with the position controller*

Table 6.11 *Ziegler-Nichols KH position controller auto-tuning method's ISE performance*

	Z-N "classic" vel. tuned with closed-loop SIMC	Z-N "classic" vel. tuned with Z-N "K-H"	Z-N "K-H" vel. tuned with closed-loop SIMC	Z-N "K-H" vel. tuned with Z-N "K-H"	Human result
ISE mean	$1.953 \cdot 10^{-3}$	$1.873 \cdot 10^{-3}$	$1.489 \cdot 10^{-3}$	$1.073 \cdot 10^{-3}$	$1.094 \cdot 10^{-2}$
ISE σ	$2.462 \cdot 10^{-4}$	$4.978 \cdot 10^{-4}$	$2.375 \cdot 10^{-4}$	$2.054 \cdot 10^{-4}$	$1.416 \cdot 10^{-2}$

As described, the KH-optimization was only run once on the target axis, with the light mass plate mounted on the axis' carriage. A new Ziegler-Nichols "K-H" variant was produced as a result. All the consecutive rounds of controller tuning and performance testing results presented here were obtained with the same optimized "K-H" variant. Based on the results, the optimized Ziegler-Nichols "K-H" variant performs well with all three mass configurations of the target system, instead of only working with the mass configuration that was used in the optimization process. Therefore, the KH-method could be suitable for creating new Ziegler-Nichols variants for new robot designs in mass production, for example. The variants could be used to produce well performing controller tuning parameters for the individual robots, regardless of their small differences in mass and friction. Based on these new results, the Ziegler-Nichols "K-H" variant is the auto-tuning method that produces the best performance for both controllers of the test system's OptoDrive.

7. DISCUSSION

The experiments conducted in this thesis have conclusively shown that automatic controller tuning is feasible for the OptoDrive. Auto-tuning methods with suitable functional behavior and good performance were found for the OptoDrive's velocity and position controllers. Two feasible auto-tuning methods were found for the velocity controller: the closed-loop SIMC method and the Ziegler-Nichols method. The Ziegler-Nichols method was found to be the only method that is feasible for auto-tuning the position controller. This is because the model-finding methods of other auto-tuning methods were found to be unsuitable for the position controller of the OptoDrive. In particular, the tested auto-tuning methods are able to produce tuning parameters on an OptoDrive-equipped system without requiring additional measurement hardware, which a desired feature of the auto-tuning implementations. The best performance on the OptoDrive was achieved with the Ziegler-Nichols method for both the velocity and the position controllers, when the Ziegler-Nichols method was optimized parameter-wise with a separate method devised during the experimentation phase of this thesis.

A novel new KH-method for adapting and optimizing the Ziegler-Nichols auto-tuning method for a new system was devised during this thesis. A new Ziegler-Nichols variant, made with the KH-method, produced highly successful results on the test axis of this thesis. According to the results, the KH-method seems to be able to abstract the mass of the system that is being auto-tuned. This is supported by the fact that the actual optimization process was only run on one test setup mass configuration, but the optimized Ziegler-Nichols variant produced high performance tuning parameters for other mass configurations as well. It remains to be determined, whether the method optimizes the Ziegler-Nichols method for a given controller, or only abstracts the mass of a given system. Considering the mathematical implementation of controllers, whether it is a parallel or ideal-form controller or some custom implementation, it would make sense that an optimization method would actually optimize the Ziegler-Nichols method for the controller. For example, some controllers use integral time in a different scale than seconds or milliseconds. Different variable scales obviously require the parameters produced by an auto-tuning method to be scaled appropriately. Additionally, some controllers operate in an arbitrary scale consisting of feedback device units, instead of metric values, and therefore require adaptation

to be compatible with auto-tuning methods in literature. The optimization method will inherently be responsible of this kind of scaling. A patenting process on the method is ongoing.

The optimized Ziegler-Nichols method surpasses the tuning performance achieved by humans by a large margin, especially considering the repeatability of tuning. On average, humans are better at tuning the velocity controller than the position controller. Humans were able to produce results closer to the auto-tuning methods for the velocity controller than the position controller. For the velocity controller, the mean tuning performance of the best automatic tuning method is about 16 percent better than humans, with only 20 percent of the standard deviation. For the position controller, the mean tuning performance of humans is one decade worse than automatic methods, with two decades worse standard deviation. It is possible that the reason behind the humans' relatively good velocity controller tuning performance and poor position controller tuning performance is that the plots produced by the drive configuration software are easier to read and interpret for the velocity controller thanks to their more favorable scale of values. From the velocity controller plots, it is easier to distinguish setpoint tracking errors in the controller's operation. In light of these results, the controller tuning done by humans can even be considered dangerous for systems that require high operational precision and repeatability, due to the high probability for humans to produce sub-optimal controller tuning parameters. These results contradict some results that were found in literature, which cite Ziegler-Nichols method's very poor performance (see Åström and Hägglund 2001, p. 1163). The method was found to perform very well in this thesis, if the formulae the method uses to calculate the resulting tuning parameters are selected correctly. This can be seen as an evidence of good general performance and repeatability of the ultimate gain experiment used by the Ziegler-Nichols method.

Future work should include characterizing the actual repeatability and measurement uncertainty of the performance measurement test that was used. It was run multiple times in both directions when testing to reduce its variance, but its repeatability should be characterized. This is especially important if the performance measurement test will be used to analyse the performance of OptoFidelity made robots in the future. This is because these kinds of performance results may end up in measurement reports. Other performance measurement indices than the ISE could also be investigated, as it is not certain that ISE is the most suitable performance index for calculating the performance of measurement robots. It would be interesting to find a performance index resembling humans' perception of controller performance. Also, additional research to find the most beneficial performance index for the actual metrology performance of OptoFidelity's robots could be completed.

Because of the controllers' coupling in OptoDrive, a new performance measurement

test could be researched. In an optimal case, the controllers could be entirely detached from each other's operation, but considering their cascade arrangement in the OptoDrive, this is not viable. However, by changing the performance measurement test, it may be possible to reduce the effect of coupling to the measurements. One possible way of doing this is to disable the position controller, and to use a separate performance test for only the velocity controller.

For future work on the auto-tuning, the KH-method could be tested with other auto-tuning methods than Ziegler-Nichols, and the resulting tuning performance compared against the KH-optimized Ziegler-Nichols. With the adjustment parameter τ_t of the setpoint overshoot and closed-loop SIMC methods, this test is straightforward. Different fitness functions and optimization algorithms could be tested for the KH-method, for searching the optimal tuning parameters. Future research should also include testing the KH-method and the produced Ziegler-Nichols "K-H" variant with different systems than a linear axis. Ball-screw driven axes, belt driven axes, and voice coil axes should be included in the testing, with different axis masses as well. After testing, a performance trigger point for re-running the KH-method, instead of simply using the existing optimized "K-H" formulae, could likely be found. The robustness of the tuning parameters resulting from the auto-tuning methods could also be researched to find a good balance between the robustness and the performance of the tuning parameters used in the OptoDrive.

The ability to auto-tune axes, enabled with the research conducted in this thesis, is an important step for OptoFidelity and a highly useful tool for future projects. For smaller batches of new robots, using an auto-tuning method allows the developers' time to be allocated to more productive tasks, eliminates visits to customer premises when the tuning values are set incorrectly by humans, and in general improves the robots' performance. The time taken to tune an axis is brought down from roughly 20 minutes or more to only a few minutes when moving from manual to automatic tuning. Auto-tuning also simplifies projects where a customer wants to use the OptoDrive system in their own robots or assemblies, as currently this requires a trained professional to visit the customer site to do the axis tuning. In a mass-production project, there can be hundreds of robots of the same type, but because of manufacturing tolerances and other variations in the assemblies, the robots require individually set tuning parameters to behave optimally. Auto-tuning therefore enables a batch of robots to be individually tuned for best performance. Additionally, some kinematic systems in robots are highly affected by the mass of a device that is put into the robot for testing or calibration. These kinds of systems have previously been tuned so that there is no oscillation in the axis when there is no device installed, leading to deterioration of robot performance when a device is put on the robot. Automatic tuning allows the robot to adapt to the mass of the

installed product, maximizing the performance and accuracy of the robot. In the best case, this increases the velocities and accelerations that can be reached with a robot, thus increasing the units per hour that can be tested. This directly leads to increased revenue that customers can receive when using OptoFidelity made robotics.

8. CONCLUSION

The research methods selected for this thesis were selected suitably, as all of the research questions were answered. The OptoFidelity OptoDrive controller can be successfully automatically tuned by using common PID controller auto-tuning methods. The results of the conducted practical experiments suggest using an optimized version of the Ziegler-Nichols controller auto-tuning method for tuning the velocity and the position controller of the OptoDrive. It utilizes a system model-identification method that has favorable behavior for a robot system, and yields superior performance to other auto-tuning methods and humans. The results also suggest that the tuning methods presented in literature are generic for PID controllers. To achieve an optimal tuning result for a given system, the auto-tuning methods should be adjusted on a case-by-case basis. The performance results show that tuning parameters found by the means of auto-tuning can be significantly more repeatable and better performing than the parameters found by humans. An optimization method for adjusting the tuning parameter calculation of existing auto-tuning methods was proposed and tested with highly successful results. For a robot project where accurate robot performance is critical, an auto-tuning method optimized with the proposed KH-method should be used instead of manual tuning.

The PID controller performance measurement methods and indices used in this thesis should be characterized more deeply, but this task is left for future consideration. The indices could be compared to human perception of robot performance and effort put into finding the most suitable performance measurement index regarding the final performance of a robot in a metrology application. The proposed method for optimization of the auto-tuning methods should be characterized and its functionality and performance for auto-tuning methods other than the Ziegler-Nichols method investigated.

The OptoDrive-optimized auto-tuning method produced in this thesis will be built into a robot control and calibration software package to be used at OptoFidelity. For OptoFidelity, the ability to auto-tune servo axis controllers means that valuable resources can be allocated into other, more productive tasks. According to the results of the conducted experiments, it will also result in better performance of the robots. Robots in production lines can be optimized individually, which allows OptoFidelity

to reach a new level of robot performance. Projects where mass installed on a robot changes during its use will see improved performance from auto-tuning, as the robot can adapt to the new mass automatically. In addition to the previously noted advantages, this increases the units per hour that a single robot can test, and therefore increases the value of the robots made by OptoFidelity.

REFERENCES

Åström, Karl J and Hägglund, Tore (1995). *PID controllers: theory, design, and tuning*. Vol. 2. Instrument society of America Research Triangle Park, NC.

Åström, Karl J and Hägglund, Tore (2001). "The future of PID control". In: *Control engineering practice* 9.11, pp. 1163–1175.

Bennett, Stuart (1993). "Development of the PID controller". In: *IEEE control systems* 13.6, pp. 58–62.

C2000™ Digital Controller Library (July 2015). SPRUI31. User's Guide. Texas Instruments.

Cervantes, Ilse and Alvarez-Ramirez, Jose (2001). "On the PID tracking control of robot manipulators". In: *Systems & control letters* 42.1, pp. 37–46.

Cheung, Norbert C (1999). "An innovative method to increase the resolution of optical encoders in motion servo systems". In: *Power Electronics and Drive Systems, 1999. PEDS'99. Proceedings of the IEEE 1999 International Conference on*. Vol. 2. IEEE, pp. 797–802.

Cohen, G. H. and Coon, G. A. (1953). "Theoretical consideration of retarded control". In: *Transactions of ASME* 75, pp. 827–834.

Grimholt, Chriss and Skogestad, Sigurd (2012). "Optimal PI-control and verification of the SIMC tuning rule". In: *IFAC Proceedings Volumes* 45.3, pp. 11–22.

Gyöngy, IJ and Clarke, DW (2006). "On the automatic tuning and adaptation of PID controllers". In: *Control engineering practice* 14.2, pp. 149–163.

Haugen, Finn (2010). "Comparing PI tuning methods in a real benchmark temperature control system". In: *Modeling, Identification and control* 31.3, p. 79.

Hjalmarsson, Håkan, Gevers, Michel, and De Bruyne, Franky (1996). "For model-based control design, closed-loop identification gives better performance". In: *Automatica* 32.12, pp. 1659–1673.

Ho, Weng Khuen, Lim, Khiang Wee, and Xu, Wen (1998). "Optimal gain and phase margin tuning for PID controllers". In: *Automatica* 34.8, pp. 1009–1014.

Koivo, Heikki and Tanttu, Juha (1991). "Tuning of PID Conrollers: Survey of Siso and Mimo Techniques". In: *Intelligent tuning and adaptive control*. Elsevier, pp. 75–80.

Lee, Chuen-Chien (1990). "Fuzzy logic in control systems: fuzzy logic controller. I". In: *IEEE Transactions on systems, man, and cybernetics* 20.2, pp. 404–418.

Lequin, Olivier et al. (2003). "Iterative feedback tuning of PID parameters: comparison with classical tuning rules". In: *Control Engineering Practice* 11.9, pp. 1023–1033.

Leva, Alberto and Maggio, Martina (2012). "Model-based PI (D) autotuning". In: *PID Control in the Third Millennium*. Springer, pp. 45–73.

Li, Hui (2002). *Rotary and linear motor*. US Patent 6,429,611.

Li, Yun, Ang, Kiam Heong, and Chong, Gregory CY (2006). "PID control system analysis and design". In: *IEEE Control Systems* 26.1, pp. 32–41.

Mahmoodabadi, MJ and Nemati, AR (2016). "A novel adaptive genetic algorithm for global optimization of mathematical test functions and real-world problems". In: *Engineering Science and Technology, an International Journal* 19.4, pp. 2002–2021.

McCormack, Anthony S and Godfrey, Keith R (1998). "Rule-based autotuning based on frequency domain identification". In: *IEEE transactions on control systems technology* 6.1, pp. 43–61.

O'Dwyer, Aidan (2009). *Handbook of PI and PID controller tuning rules*. Imperial college press.

Ogata, Katsuhiko and Yang, Yanjuan (2002). *Modern control engineering*. Vol. 4. Prentice hall India.

Pessen, David W (1994). "A new look at PID-controller tuning". In: *Journal of dynamic systems, measurement, and control* 116.3, pp. 553–557.

Pillay, Pragasen and Krishnan, Ramu (1991). "Application characteristics of permanent magnet synchronous and brushless DC motors for servo drives". In: *IEEE Transactions on industry applications* 27.5, pp. 986–996.

Ponton, Jack W. (2007). *ECOSSE Control HyperCourse Module 4: Further Controller Design*. `http://homepages.ed.ac.uk/jwp/control06/controlcourse/restricted/course/advanced/module4.html`. Accessed: 20.05.2018.

Porter, Brian and Jones, AH (1992). "Genetic tuning of digital PID controllers". In: *Electronics letters* 28.9, pp. 843–844.

Raunt, KH and Vaishnay, SR (2012). "A Study on Performance of Different PID Tuning Techniques". In: *International Conference on Electrical Engineering and Computer Science*, pp. 250–254.

Richalet, J et al. (1987). "Predictive functional control-application to fast and accurate robots". In: *IFAC Proceedings Volumes* 20.5, pp. 251–258.

Rivera, Daniel E, Morari, Manfred, and Skogestad, Sigurd (1986). "Internal model control: PID controller design". In: *Industrial & engineering chemistry process design and development* 25.1, pp. 252–265.

Robinson, Jacob and Rahmat-Samii, Yahya (2004). "Particle swarm optimization in electromagnetics". In: *IEEE transactions on antennas and propagation* 52.2, pp. 397–407.

Scipy (2017). *Discrete Fourier Transform*. `https://docs.scipy.org/doc/numpy-1.13.0/reference/routines.fft.html`. Accessed: 26.07.2018.

Shamsuzzoha, Mohammad and Skogestad, Sigurd (2010). "The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning". In: *Journal of Process Control* 20.10, pp. 1220–1234.

Shieh, Yaw-Shih, Lee, An-Chen, and Chen, Chin-Sheng (1996). "Cross-coupled biaxial step control for CNC EDM". In: *International Journal of Machine Tools and Manufacture* 36.12, pp. 1363–1383.

Shinskey, FG (1990). "How good are our controllers in absolute performance and robustness?" In: *Measurement and Control* 23.4, pp. 114–121.

Silva, Guillermo J, Datta, Aniruddha, and Bhattacharyya, Shankar P (2007). *PID controllers for time-delay systems*. Springer Science & Business Media.

Skogestad, Sigurd (2003). "Simple analytic rules for model reduction and PID controller tuning". In: *Journal of process control* 13.4, pp. 291–309.

Skogestad, Sigurd and Grimholt, Chriss (2012). "The SIMC method for smooth PID controller tuning". In: *PID Control in the Third Millennium*. Springer, pp. 147–175.

Smuts, Jacques (2011). *Cohen-Coon Tuning Rules*. `http://blog.opticontrols.com/archives/383`. Accessed: 13.07.2018.

Srinivas, Mandavilli and Patnaik, Lalit M (1994). "Adaptive probabilities of crossover and mutation in genetic algorithms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.4, pp. 656–667.

Tan, Wen et al. (2006). "Comparison of some well-known PID tuning formulas". In: *Computers & chemical engineering* 30.9, pp. 1416–1423.

Veronesi, Massimiliano and Visioli, Antonio (2010). "Performance assessment and retuning of PID controllers for integral processes". In: *Journal of Process Control* 20.3, pp. 261–269.

Vilanova, Ramon and Visioli, Antonio (2012). *PID control in the third millennium*. Springer.

Visioli, Antonio (1999). "Fuzzy logic based set-point weight tuning of PID controllers". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: systems and humans* 29.6, pp. 587–592.

Visioli, Antonio (2006). *Practical PID Control*. Advances in Industrial Control. Springer London. ISBN: 9781846285868. URL: `https://books.google.fi/books?id=ymyAY01bEe0C`.

Ziegler, John G and Nichols, Nathaniel B (1942). "Optimum settings for automatic controllers". In: *trans. ASME* 64.11.

APPENDIX A: TEST SYSTEM HARDWARE BREAKDOWN

Component	Vendor	Model
Linear motor forcer	Chieftec	LM-PA-X2
Magnet track	Chieftec	LM-SA-X1
Linear guide	Hiwin	WER27R640P
Linear bearings	Hiwin	WEH27CA
Optical scale	RSF	AK MS15 version MK
Optical read head	RSF	AK MS15 TTLx100
Servo drive	OptoFidelity	OptoDrive
Power supply	ProPower	PS3003

APPENDIX B: MANUAL TESTING PLAN

Test phase	Component	Description
Introduction	Purpose of test	Explain the purpose of the test
	System overview	Introduce the testers to the system
Walk-through	Test flow	Describe the test flow to the testers
	Test output	List required parameters to tune
Practical phase	Tuning w/out a mass plate	Tune axis and report parameters
	Tuning w/ light mass plate	
	Tuning w/ heavy mass plate	
Feedback	Feedback to testers	Give feedback of how testing went
	Feedback to organizers	Gather feedback about test

APPENDIX C: TUNING PARAMETERS FROM THE HUMAN PID TUNING TEST

	Tester	Vel K_p	Vel K_i	Pos K_p	Vel ISE	Pos ISE
No mass plate	1	202	104	498	395.7	0.05503
	2	250	200	2000	196.0	0.00226
	3	250	350	1110	151.1	0.00209
	4	350	180	800	174.2	0.00908
	5	250	350	100	117.7	0.02107
Light mass	1	235	157	768	330.2	0.02095
	2	350	350	2000	187.7	0.00133
	3	440	455	790	103.8	0.00278
	4	350	200	1600	216.2	0.00445
	5	325	500	200	115.7	0.00996
Heavy mass	1	307	198	801	278.6	0.01658
	2	400	350	2500	188.4	0.00110
	3	480	550	1200	136.8	0.00144
	4	350	300	1900	201.4	0.00225
	5	325	500	150	119.8	0.01375

APPENDIX D: COMBINING PERFORMANCE RESULTS OF DIFFERENT MASS CONFIGURATIONS

The controller performance measurement test, outlined in Section 4.1, produces controller tracking error data. An ISE performance index can be calculated from the data. The controller auto-tuning method testing in this thesis is conducted with the three mass configurations that are available for the test axis (see Section 3.1), with ten repetitions for each of the configurations. This results in a set of 30 ISE indices for each of the tested auto-tuning methods. From the perspective of handling the performance results and comparing the results of several auto-tuning methods, it is beneficial if the performance results for each of the mass configurations can be combined. Combining the results is theoretically possible, because the mass on the axis does not affect the ideal trajectory that is being used during the test, and because the auto-tuning method on hand should adapt the target PID controller to operate with the current mass configuration on the system. Therefore, the resulting ISE indices should describe how well the auto-tuning method worked with the current mass configuration, without additional magnitude resulting from the configuration.

This theory was tested by analysing the ISE values produced with the best velocity controller auto-tuning methods that were tried in this thesis. By comparing the best ISE results for each of the axis mass configurations, the auto-tuning methods' effect on the results can be minimized. According to the ISE results, shown in the Appendix E, the best ISE results for the mass configurations were produced by the closed-loop SIMC and the Ziegler-Nichols "K-H" methods. The closed-loop SIMC method produced the best ISE results for the light mass plate configuration, and the Ziegler-Nichols "K-H" variant produced the best ISE values for the no mass plate and the heavy mass plate configurations. The tuning parameters and ISE results for these three cases were gathered to Table D.1, along with the mean values for the ISE results.

The closed-loop SIMC method achieved almost an equal mean ISE result with the light mass plate, than the Ziegler-Nichols "K-H" achieved with the heavy mass plate. The ISE results of the Ziegler-Nichols "K-H" variant for the no mass plate configuration are slightly worse than the other results. This can be caused by resonances, behavior of axis friction, and sampling related errors, among others. However, as the ISE index calculates a square of the controller's tracking error, the real-world difference between the results is not very significant. The magnitudes of the performance results are comparable, and do not consistently increase or decrease with added mass. Similar results can be found when analysing the best ISE values

Table D.1 *The best achieved tuning parameters and ISE results for the velocity controller*

No mass plate Z-N KH			Light mass plate Closed-loop SIMC			Heavy mass plate Z-N KH		
K_p	K_i	ISE	K_p	K_i	ISE	K_p	K_i	ISE
270	207	166.7	294	516	143.0	474	382	150.7
264	204	176.6	292	514	145.4	516	411	175.9
263	204	178.8	294	518	152.4	480	387	153.9
259	202	192.7	294	522	144.5	468	378	139.7
259	203	197.0	292	484	155.0	504	401	150.9
266	208	164.1	292	510	173.5	492	397	145.8
261	203	185.9	280	496	171.9	469	378	174.8
269	207	165.9	296	524	147.9	491	397	138.8
258	201	189.4	292	516	150.0	503	401	172.5
276	211	150.0	294	478	137.8	482	388	133.9
Mean ISE		176.7			152.1			153.7

for the position controller (see Appendix F, test 5).[1]

Based on the test, the mass and the magnitude of the ISE results do not directly correlate. The almost equal performance results for the different mass configurations show that similar performance results can be expected regardless of mass, provided that an auto-tuning method functions well with the system. This is a justification for combining all the ISE results of a given auto-tuning method. Therefore, the different–mass ISE results of an auto-tuning method can be considered as a single set, from which statistical indices can be calculated.

[1]The ISE results for the different mass configurations are very similar, however with slightly worse results for the light mass plate configuration. The results do not consistently increase or decrease with added mass.

APPENDIX E: VELOCITY CONTROLLER TUNING PARAMETERS AND PERFORMANCE RESULTS

Table E.1 *Legend for the velocity controller data*

	Velocity controller tuning method
Test 1	Setpoint overshoot
Test 2	Closed-loop SIMC
Test 3	Ziegler-Nichols "some overshoot"
Test 4	Ziegler-Nichols KH

Table E.2 *Velocity controller tuning parameters and performance results*

	Test 1			Test 2			Test 3			Test 4		
	K_p	K_i	ISE	K_p	K_i	ISE	K_p	K_i	ISE	K_p	K_i	ISE
No mass plate	145	283	424.0	192	340	224.6	342	308	268.4	270	207	166.7
	150	312	441.1	194	344	200.8	302	264	332.7	264	204	176.6
	143	270	531.1	192	338	215.4	306	256	343.8	263	204	178.8
	144	275	444.6	202	316	226.2	290	252	345.8	259	202	192.7
	146	322	380.4	168	258	294.0	304	264	346.7	259	203	197.0
	146	318	435.5	196	348	258.8	302	258	370.3	266	208	164.1
	143	281	485.0	192	338	198.8	276	242	390.1	261	203	185.9
	144	283	434.1	194	342	266.0	298	268	370.9	269	207	165.9
	146	316	377.3	192	340	205.7	308	258	359.7	258	201	189.4
	145	329	417.0	172	266	311.2	300	260	375.2	276	211	150.0
Light mass plate	208	343	530.9	294	516	143.0	474	414	289.6	432	345	184.0
	206	347	510.4	292	514	145.4	462	404	299.3	430	340	160.4
	225	295	479.4	294	518	152.4	464	412	289.6	408	328	163.4
	234	291	442.5	294	522	144.5	468	412	313.9	420	340	150.2
	227	287	495.8	292	484	155.0	470	416	271.2	435	339	184.4
	222	285	485.9	292	510	173.5	462	414	283.6	420	338	162.1
	197	362	536.1	280	496	171.9	434	392	315.8	409	328	164.2
	227	285	468.6	296	524	147.9	462	416	314.7	428	339	141.7
	230	287	472.5	292	516	150.0	464	424	272.6	427	339	164.3
	224	275	503.5	294	478	137.8	448	402	303.9	423	334	156.7
Heavy mass plate	261	315	673.4	324	572	165.8	528	474	561.5	474	382	150.7
	268	252	686.2	324	564	160.1	554	498	501.2	516	411	175.9
	269	264	710.9	324	566	167.4	566	502	558.8	480	387	153.9
	270	248	682.5	326	574	156.4	528	472	500.1	468	378	139.7
	255	208	829.0	324	574	184.2	514	462	535.7	504	401	150.9
	274	250	665.1	326	570	136.7	554	490	481.6	492	397	145.8
	273	223	753.9	328	574	152.7	552	484	470.5	469	378	174.8
	285	262	589.9	328	574	134.0	514	460	611.4	491	397	138.8
	274	252	655.3	330	578	136.8	524	474	576.6	503	401	172.5
	266	243	704.6	328	570	144.5	530	474	547.1	482	388	133.9

Position controller K_p of 1800 was used during all the measurements

APPENDIX F: POSITION CONTROLLER TUNING PARAMETERS AND PERFORMANCE RESULTS

Table F.1 *Legend for the position controller data*

	Position controller tuning method	Velocity controller tuning method [1]
Test 1	Ziegler-Nichols "classic"	Closed-loop SIMC
Test 2	Ziegler-Nichols "classic"	Ziegler-Nichols "some-overshoot"
Test 3	Ziegler-Nichols "classic"	Ziegler-Nichols KH
Test 4	Ziegler-Nichols KH	Closed-loop SIMC
Test 5	Ziegler-Nichols KH	Ziegler-Nichols KH

Table F.2 *Position controller tuning parameters and performance results*

	Test 1		Test 2		Test 3		Test 4		Test 5	
	K_p	ISE	K_p	ISE	K_p	ISE	K_p	ISE	K_p	ISE
No mass plate	2671	0.002168	2861	0.003057	1262	0.003005	3551	0.001545	1945	0.000898
	2599	0.002216	2904	0.003129	1421	0.002506	3636	0.001534	1877	0.001014
	2629	0.002228	2906	0.003010	1559	0.002171	3553	0.001731	1944	0.000929
	2676	0.002151	2901	0.003254	1488	0.002176	3514	0.001725	2090	0.000840
	2594	0.002352	2959	0.003237	1260	0.003106	3594	0.002031	1946	0.000932
	2645	0.002419	2910	0.003309	1523	0.002252	3636	0.001738	1874	0.001015
	2644	0.002240	2900	0.003318	1453	0.002447	3607	0.001871	2017	0.000903
	2585	0.002349	2922	0.003449	1359	0.002802	3534	0.001910	1942	0.000970
	2676	0.002352	2950	0.003270	1487	0.002507	3669	0.002076	1946	0.000937
	2604	0.002403	2926	0.003634	1541	0.002397	3605	0.001704	2087	0.000884
Light mass plate	2302	0.001864	2600	0.002330	1787	0.001652	3176	0.001506	2343	0.001396
	2321	0.001886	2543	0.002608	1874	0.001557	3153	0.001403	2339	0.001341
	2389	0.001714	2602	0.002460	1850	0.001615	3215	0.001324	2259	0.001452
	2342	0.001764	2641	0.002341	1767	0.001702	3110	0.001451	2613	0.001177
	2330	0.001764	2581	0.002399	1637	0.001917	3139	0.001484	2432	0.001332
	2432	0.001746	2590	0.002437	1872	0.001526	3246	0.001186	2340	0.001407
	2400	0.001654	2610	0.002599	1748	0.001781	3199	0.001294	2431	0.001302
	2337	0.001845	2581	0.002400	1871	0.001538	3132	0.001337	2613	0.001293
	2421	0.001635	2562	0.002634	1620	0.001845	3254	0.001244	2256	0.001458
	2384	0.001715	2631	0.002475	1509	0.002139	3271	0.001366	2343	0.001433
Heavy mass plate	2376	0.001814	2479	0.002301	1720	0.001462	3156	0.001333	2207	0.000915
	2411	0.001878	2440	0.002441	1721	0.001492	3264	0.001241	2286	0.000936
	2393	0.001878	2512	0.002313	1602	0.001642	3199	0.001547	2057	0.001063
	2353	0.001916	2482	0.002393	1654	0.001587	3153	0.001396	2217	0.000949
	2435	0.001773	2455	0.002454	1749	0.001398	3239	0.001388	2206	0.000925
	2397	0.001820	2470	0.002405	1689	0.001515	3199	0.001379	2291	0.001008
	2372	0.001900	2500	0.002251	1602	0.001638	3161	0.001353	2206	0.000980
	2399	0.001899	2460	0.002249	1753	0.001386	3263	0.001402	2283	0.000959
	2423	0.001841	2449	0.002361	1659	0.001470	3254	0.001365	2058	0.001100
	2388	0.001839	2522	0.002277	1600	0.001618	3169	0.001317	2289	0.000965

[1] Best controller tuning parameters based on the ISE results were used (see Appendix E)